ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC MÁY TÍNH

NGUYỄN TRƯỜNG PHÁT

KHÓA LUẬN TỐT NGHIỆP

KẾT HỢP ẢNH VÀ CÂU MÔ TẢ TĂNG CƯỜNG TIẾNG VIỆT CHO TRUY VẤN ẢNH

CỬ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

TP. HỒ CHÍ MINH, 2021

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC MÁY TÍNH

NGUYỄN TRƯỜNG PHÁT – 17520880

KHÓA LUẬN TỐT NGHIỆP

KẾT HỢP ẢNH VÀ CÂU MÔ TẢ TĂNG CƯỜNG TIẾNG VIỆT CHO TRUY VẤN ẢNH

CỬ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

GIẢNG VIÊN HƯỚNG DẪN

TS. NGUYỄN VINH TIỆP

TP. HỒ CHÍ MINH, 2021

DANH SÁCH HỘI ĐỒNG BẢO VỆ KHÓA LUẬN

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số …… ngày …….. của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

1. ……. – Chủ tịch.
2. ……. – Thư ký.
3. ……. – Ủy viên.
4. ……. – Ủy viên.

LỜI CẢM ƠN

Khoảng thời gian làm khóa luận chắc chắn là một trong những khoảng thời gian đáng nhớ nhất trong cuộc đời em, thật nhiều cảm xúc.

Lời đầu tiên em xin dành lời cảm ơn đặc biệt đến với thầy T.S Nguyễn Vinh Tiệp đã hướng dẫn và dìu dắt em trong suốt quá trình thực hiện luận văn tốt nghiệp. Thầy như một người cha, người bạn và một người anh em tràn đầy năng lượng. Thầy sẵn sàng góp ý về mọi thứ , nhờ đó mà em trưởng thành lên rất nhiều. Chưa bao giờ em gặp một người thầy cởi mở với em đến như vậy.

Em xin cảm ơn thầy T.S Lê Đình Duy đã đưa ra những lời góp ý sâu sắc, giúp em hiểu ra thêm nhiều vấn đề khi trình bày thuyết trình khóa luận. Nhờ thầy góp ý, phần trình bày em được cô đọng và súc tích hơn trước, dù chỉ gặp thầy trong khoảng thời gian rất ngắn ngủi.

Em xin cảm ơn thầy phản biện TS. Đinh Quang Vinh từ trường Đại học Việt Đức (VGU), thầy đã có những góp ý rất sâu sắc về cách trình bày, cách tạo điểm nhấn trong bài trình bày. Thầy cũng đưa ra những ý tưởng và truyền cho em những động lực nghiên cứu trong tương lai.

Em xin cảm ơn Phòng thí nghiệm Truyền thông Đa phương tiện MMLab đã tạo một môi trường cho các bạn sinh viên và em có thể nghiên cứu học hỏi. Qua thời gian ở lab em đã học ra được rất nhiều thứ. Cảm ơn những người anh: anh Nguyễn Nhật Duy, anh Nguyễn Minh Dũng và anh Nguyễn Vũ Anh Khoa đã đưa ra những góp ý và giúp đỡ em chỉnh chu hơn trong việc làm khóa luận. Cảm ơn những người bạn đồng hành: Vũ Đình Vi Nghiệm, Lê Thanh Phước Hiếu, Lê Hoàng Ân, Nguyễn Hoàng Trung, Hồ Sỹ Tuyến, Nguyễn Thành Danh, Phan Nguyên và Đặng Hoàng Sang.

Con cảm ơn gia đình đã luôn là chỗ dựa tinh thần, luôn kề vai sát cánh, ủng hộ con trên con đường mà con đã chọn.

Cảm ơn anh Bùi Lê Duy Nhất và anh Hoàng Hữu Tín ở Cinnamon AI dõi theo, tạo động lực và góp ý cho những ý tưởng của em. Cảm ơn Trần Vinh Hưng, Nguyễn Trọng Tùng, Phạm Hồng Vinh, Bùi Thị Cẩm Nhung và Lê Tấn Đăng Tâm là những người bạn ở

trường Đại học Khoa học Tự nhiên đã đồng hành cùng trong suốt quá trình em thực hiện khóa luận tốt nghiệp.

MỤC LỤC

DANH SÁCH HÌNH VẼ

DANH MỤC BẢNG

DANH MỤC TỪ VIẾT TẮT

CNN	Convolutional Neural Network
LSTM	Long Short-term Memory
TIRG	Text-Image Residual Gating
MLP	Multilayer Perceptron
XLNNTN	Xử lý ngôn ngữ tự nhiên
TTNT	Trí tuệ nhận tạo
MNNT	Mạng nơ-ron nhân tạo
TVTT	Truy vấn thông tin

TÓM TẮT KHÓA LUẬN

Truy vấn ảnh sử dụng kết hợp ảnh và câu mô tả tăng cường là một bài toán truy vấn ảnh dựa trên một tấm ảnh tham khảo cho trước, với một số thay đổi mong muốn của người dưới dạng một câu mô tả tăng cường ở dạng ngôn ngữ tự nhiên. Nhìn chung, khi truy vấn, người dùng đã hình dung thứ mà họ muốn trong đầu, tuy nhiên họ chưa biết cách nào để truyền tải cái họ muốn vào hệ thống tìm kiếm một cách hiệu quả. Việc cho phép người dùng sử dụng một tấm ảnh họ đã có sẵn kèm với một câu mô tả tăng cường giúp họ có thể thoải mái và linh hoạt hơn trong việc truyền tải nhu cầu thông tin vào trong hệ thống tìm kiếm. Đây là một bài toán có rất nhiều tiềm năng ứng dụng trong cuộc sống nhờ tính thuận tiện trong việc mô tả câu truy vấn, tuy nhiên chưa được khai thác trên ngôn ngữ tiếng Việt. Trong khóa luận này, chúng tôi tập trung nghiên cứu một phương pháp biểu diễn hiệu quả cho cặp ảnh và câu mô tả tăng cường tiếng Việt, để có thể sử dụng biểu diễn này để thực hiện truy vấn trong cơ sở dữ liệu ảnh. Khóa luận tập trung nghiên cứu phương pháp **Text Image Residual Gating** được đề xuất ở hội nghị **CVPR2019**. Trong đó, chúng tôi nghiên cứu, tìm hiểu, thực nghiệm và đánh giá phương pháp được đề xuất ở bài báo trên, đồng thời cũng xây dựng tập dữ liệu tiếng Việt để kiểm tra tính khả thi của phương pháp này với dữ liệu tiếng Việt. Thông qua đó, chúng tôi cũng xây dựng được một bộ công cụ dịch tự động dựa trên cây cú pháp có tính hiệu quả cao và tốn ít tài nguyên. Để giúp mô hình thích ứng được với các câu mô tả tăng cường có từ nằm ngoài từ điển của tập huấn luyện, chúng tôi sử dụng **RDRSegmenter** cho bộ tách từ và một mô hình học máy tiền huấn luyện là **PhoBERT** cho việc biểu diễn từ thay thế và đạt được kết quả tốt. Qua đó, chúng tôi cũng rút trích ra được rất nhiều bài học và góc nhìn rất thú vị để phục vụ cho nghiên cứu sắp tới.

Chương 1. TỔNG QUAN

Trong chương này chúng tôi giới thiệu tổng quan về bài toán truy vấn ảnh sử dụng kết hợp ảnh và câu mô tả tăng cường, tình hình nghiên cứu của các bài toán liên quan, đồng thời chia sẻ ngắn gọn mục tiêu và kết quả nghiên cứu của khóa luận.

1.1. Giới thiệu bài toán

Bài toán truy vấn ảnh là bài toán đã có mặt từ rất lâu đời, kể từ khi khoa học máy tính vừa phát triển thì việc truy vấn đã trở thành một đề tài nghiên cứu được chú ý. Ngày nay với sự phát triển không ngừng của internet, dữ liệu được đăng tải mỗi ngày trên các trang mạng xã hội đạt số lượng tới số lượng khổng lồ. Do đó, nhu cầu tìm kiếm trở nên quan trọng hơn bao giờ hết.

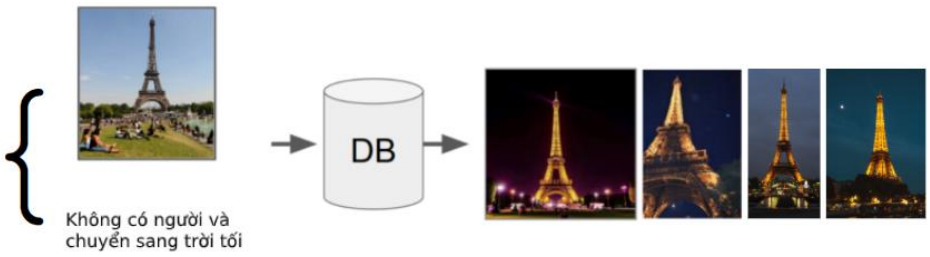
Đề tài của chúng tôi là bài toán truy vấn ảnh sử dụng kết hợp ảnh và câu mô tả tăng cường, với:

Đầu vào là:

- Tấm ảnh tham khảo và câu mô tả tăng cường tiếng Việt
- Cơ sở dữ liệu ảnh

Đầu ra là:

- Danh sách các ảnh được sắp xếp theo độ phù hợp giảm dần



Ảnh 1.1. Ảnh minh họa bài toán kết hợp ảnh và câu mô tả tăng cường cho truy vấn ảnh

Việc kết hợp cả ảnh và câu mô tả tăng cường cho phép người diễn đạt chặt chẽ ý định tiềm kiếm của mình, giảm khoảng cách ý định (intention gap). Đồng thời, văn bản là một giao thức đơn giản và linh hoạt để người dùng có thể giao tiếp nhu cầu thông tin của mình cho hệ thống tìm kiếm. Hơn hết, người dung có thể tận dụng được ảnh tham khảo sẵn có, kém với một số thay đổi mong muốn, được biểu diễn ở dạng ngôn ngữ tự nhiên.

Có thể thấy, đây là một bài toán mới và có rất nhiều tiềm năng ứng dụng trong tương lai. Tuy nhiên chưa có công trình nào nghiên cứu về phương pháp này cho ngôn ngữ tiếng Việt, tức đối với câu mô tả là tiếng Việt, với những đặc thù về ngôn ngữ rất riêng.

Có thể thấy, **Tiếng Việt** sở hữu một lượng từ phân loại (categorical nouns/classifiers) rất phong phú [43]. Đây là những từ được cho là trợ từ (helper words) cho những từ đằng sau nó, là một đặc điểm giúp mô hình Học máy mô hình hiệu quả ở trên tiếng Việt.

Từ phân loại	Ý nghĩa	V.dụ ở tiếng Việt	V.dụ ở tiếng Anh
Con	Chỉ động vật	Một **con** mèo	A cat
Quyển	Chỉ vật giống sách	Một **quyển** sổ	A notebook
Cái	Chỉ vật thể	Một **cái** bàn	A table

Bảng 1.1. Ví dụ cho từ phân loại ở tiếng Việt

Những từ này cho thêm thông tin về danh từ đứng phía sau nó, điều này rất thuận tiện khi mô hình hóa trên các phương pháp Học máy.

Ngoài ra tiếng Việt với đặc tính là **không có biến tố** (non-inflection) [43], sẽ tiềm năng khi mô hình hóa bằng mô hình học máy [42, 44]. Ở một số ngôn ngữ có biến tố như tiếng Anh hay tiếng Đức, các từ sẽ được thêm một thành phần phụ tố để thỏa mãn quy tắc ngữ pháp của ngôn ngữ đó. Ví dụ ở tiếng Anh, động từ “work” (đi làm) khi đại từ nhân xưng “he” (anh ấy) sẽ phải thêm một phụ tố “s” vào thành

từ “works” trong khi “works” và “work” là hai từ tương đồng nhau về mặt ngữ nghĩa, tiếng Việt không tồn tại hiện tượng này. Mặt khác, khi biểu diễn động từ trong quá khứ, tiếng Anh sẽ thêm hậu tố “-ed” vào động từ đó, ví dụ như “worked” (đã làm) hay “studied” (đã học). Thay vào đó, tiếng Việt thêm một phụ từ “đã” ở trước động từ để diễn đạt một hành động đã được xảy ra trong quá khứ, việc này sẽ giúp cung cấp thêm thông tin cho mô hình học máy trên các mô hình hồi quy, giúp việc mô hình hóa ngôn ngữ tiếng Việt hiệu quả hơn.

Tuy vậy, ngôn ngữ tiếng Việt vẫn tồn đọng sự **nhập nhằng khoảng trắng**. Ở tiếng Anh, khoảng trắng được sử dụng để ngăn cách giữa các từ trong một câu, còn ở tiếng Việt thì chỉ để ngăn cách giữa các âm tiết với nhau. Ví dụ, những từ như “quần áo” hay “sách vở” là một từ nhưng lại bị ngăn cách bởi một khoảng trắng. Hơn nữa, đối với một số từ láy như “thăm thẳm” hay “dào dạt”, các âm tiết được ngăn cách bởi khoảng cách trên không thể tạo thành một đơn vị ngữ nghĩa. Theo, có khoảng 85% từ tiếng Việt được cấu thành bởi hai âm tiết và có hơn 80% các âm tiết bản thân nó đã là một từ [60]. Điều này làm bài toán tách từ ở tiếng Việt trở thành một bài toán khó và đầy thử thách.

Những lợi thế của ngôn ngữ tiếng Việt hứa hẹn một tiềm năng nghiên cứu của bài toán truy vấn ảnh sử dụng ảnh và câu mô tả tăng cường đối với tiếng Việt. Việc tồn đọng những khó khăn trong mô hình hóa tiếng Việt cũng là một cơ hội để khai thác và cải tiến phương pháp.

1.2. Tình hình nghiên cứu của các bài toán liên quan

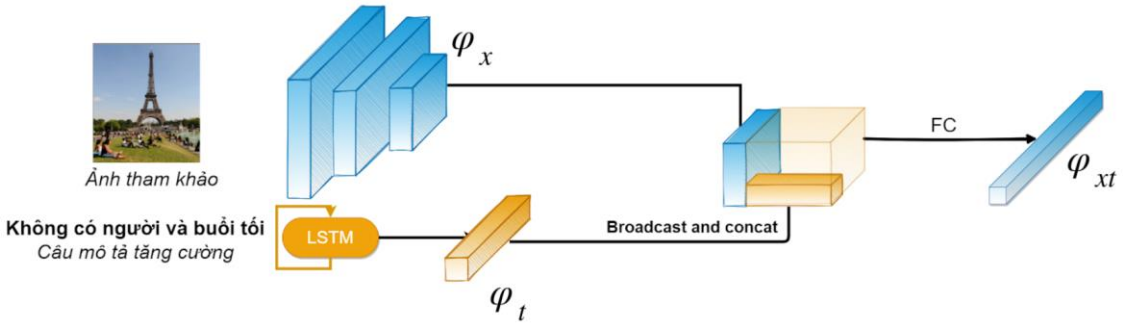
Bài toán truy vấn ảnh là một bài toán lâu đời, tuy nhiên giao thức biểu diễn truy vấn (query) dựa trên ảnh và câu mô tả là một giao thức còn rất mới. Tuy nhên, các phương pháp kết hợp đặc trưng ảnh và văn bản đã có mặt từ khá lâu và cũng đã có một số tiến triển nhất định, là một trong phương pháp khả dĩ để kết hợp biểu diễn ảnh và câu mô tả cho truy vấn. Cụ thể là trong bài toán **Hỏi đáp trên ảnh** (*Visual Question Answering*), hệ thống nhận đầu vào là ảnh và một câu hỏi

dưới dạng ngôn ngữ tự nhiên. Ngoài ra, chúng ta sẽ điểm qua tình hình của các bài toán Xử lí ngôn ngữ tự nhiên trên tiếng Việt cũng như Học hỗn hợp.

1.2.1. Hỏi đáp trên ảnh

Các phương pháp kết hợp biểu diễn ảnh và biểu diễn của văn bản thành đã có có một số tiến triển nhất định và có nhiều ứng dụng trong nhiều lĩnh vực, đặc biệt là **Hỏi đáp trên ảnh**. Bài toán Hỏi đáp trên ảnh nhận được rất nhiều sự chú ý trong thời gian gần đây. Rất nhiều phương pháp kết hợp biểu diễn vectơ của ảnh ϕ_x và văn bản ϕ_t thành phép biểu diễn kết hợp cho cặp (ảnh, văn bản) là ϕ_{xt} một cách hiệu quả được đề xuất. Nhìn chung, các phương pháp này đều nhắm đến cách xây dựng một loại đặc trưng *"hoàn toàn mới"*, không nằm trong không gian của ảnh ban đầu, do mục tiêu xây dựng các đặc trưng này là để giải quyết bài toán Hỏi đáp trên ảnh chứ không trực tiếp giải quyết bài toán truy vấn ảnh.

Phương pháp concatenation được sử dụng phổ biến để đưa biểu diễn ảnh ϕ_x và văn bản ϕ_t về một *không gian chung*, ta gọi phép biểu diễn này là ϕ_{xt}. Phương pháp này tuy đơn giản nhưng được chứng minh tính hiệu quả trong rất nhiều ứng dụng [10, 11, 12,13].



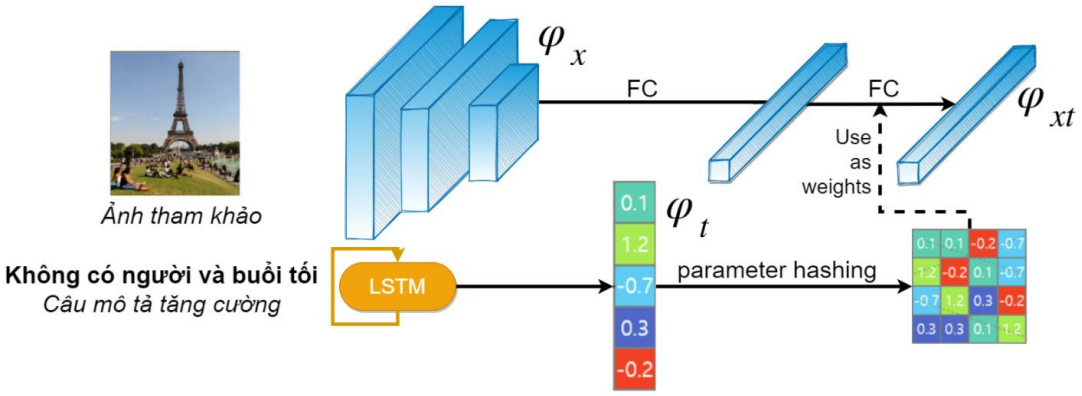
Ảnh 1.2. Phương pháp concatenation cho dung nạp ảnh và văn bản

Show and Tell [30] sử dụng mạng **LSTM** để mã hóa cặp (ảnh, văn bản) bằng cách cho bản đồ đặc trưng vào những thời điểm (time step) đầu tiên của LSTM,

theo sau bởi cách vectơ đặc trưng của các từ trong câu (một cách tuần tự). Biểu diễn sinh ra ở thời điểm cuối cùng được dùng làm ϕ_{xt}.

Relationship [32] sử dụng CNN để rút trích bản đồ đặc trưng ảnh ϕ_x, sau đó tạo một tập các đặc trưng liên quan đến nhau, mỗi đặc trưng này bao gồm viết chồng (concatenate) đặc trưng văn bản ϕ_t và 2 đặc trưng cục bộ của ϕ_x. Tập này được đưa vào một mạng lan truyền thuận đa tầng và kết quả được lấy trung bình để lấy biểu diễn kết hợp cho ảnh và văn bản ϕ_{xt}

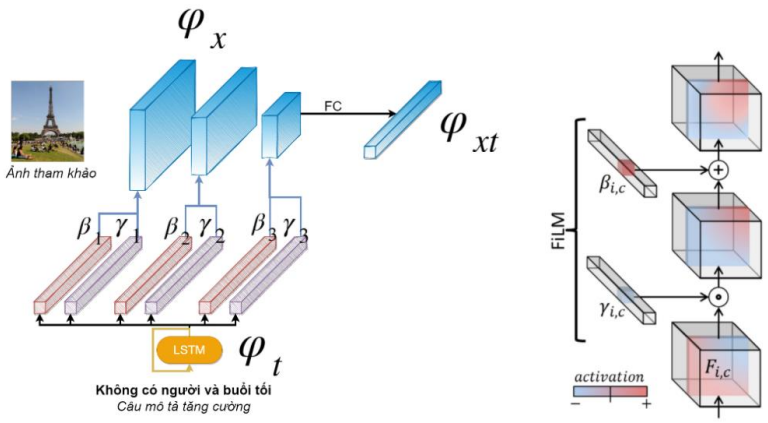
Parameter Hashing [33] là một kĩ thuật thường được sử dụng cho bài toán Hỏi đáp trên ảnh. Biểu diễn văn bản ϕ_t sẽ được băm thành một ma trận biến đổi T_t, để sau đó được nhân với bản đồ đặc trưng ảnh ϕ_x, dùng để thay thế cho lớp kết nối đầy đủ trong mạng CNN. Kết quả của phép biến đổi sẽ được sử dụng cho biểu diễn cặp ảnh và văn bản ϕ_{xt}.



Ảnh 1.3. Phương pháp parameter hashing cho dung nạp văn bản

Phương pháp gần nhất với phương pháp chúng tôi là **FiLM** [31], những đặc trưng văn bản được nhúng kết hợp với đặc trưng ảnh bằng cách sử dụng các đặc trưng này như một bộ tham số cho lớp Fully Connected của một mạng CNN. Phương pháp này có vẻ rất giống với phương pháp của chúng tôi đang sử dụng, tuy nhiên lại khác ở những điểm quan trọng cốt yếu:

1) Những biến đổi đặc trưng ở phương pháp chúng tôi được học bằng việc sử dụng cả đặc trưng văn bản và ảnh, thay vì chỉ sử dụng đặc trưng văn bản đơn thuần.
2) Phương pháp **TIRG** chúng tôi sử dụng có các phép biến đổi phi tuyến và sử dụng *nhiều tham số hơn*, so với những phép biến đổi tuyến tính và ít tham số của **FiLM**. Đó là lý do tại sao lớp **FiLM** chỉ có thể thực hiện những toán tử cơ bản như *phép tỉ lệ* (scaling), *phép phủ định* (negating) và *phép lấy ngưỡng* (thresholding)
3) Vì chỉ thực hiện các toán tử cơ bản nên **FiLM** cần phải được nhúng vào tất cả các lớp để có thể thực hiện các toán tử phức tạp còn **TIRG** chỉ được thực hiện trên một lớp của mạng. Điều này rất quan trọng để đảm bảo đặc trưng biến đổi này nằm trong không gian biểu diễn của ảnh mục tiêu.



Ảnh 1.4. Phương pháp FiLM cho dung nạp ảnh và văn bản

1.2.2. Các bài toán Xử lí ngôn ngữ tự nhiên trên tiếng Việt

Các mô hình Học máy gần đây đã tạo được rất nhiều tiếng vang do tính ứng dụng và hữu ích cao của chúng. Mặc dù vậy, trước đây, những mô hình học máy chưa được khai thác nhiều trên ngôn ngữ tiếng Việt do tính địa phương của bài toán mô hình hóa ngôn ngữ, một mô hình được huấn luyện trên dữ liệu tiếng Anh sẽ không thể hoạt động được trên dữ liệu tiếng Việt và ngược lại.

Gần đây, rất nhiều phương pháp giải quyết các bài toán Xử lí ngôn ngữ tự nhiên trên tiếng Việt ra đời. Ví dụ điển hình các bài toán **Dịch máy** [46, 47, 48], **Phân tích cảm xúc** [49, 50, 51] hay **Sinh ngữ** [52]. Những bộ dữ liệu benchmark tiếng Việt cũng đã được ra đời để huấn luyện và đánh giá các mô hình Học Máy [53, 54]. Để giải quyết vấn đề nhập nhằng khoảng trắng ở tiếng Việt, một số mô hình tách từ được ra đời [58, 59], và gần đây nhất là **RDRSegmenter** [57], đánh bại tất cả các mô hình tách từ state-of-the-art trước đó. Năm 2020 mô hình ngôn ngữ tiền huấn luyện **PhoBERT** [22] ra đời đặt nền móng cho các ứng dụng Học máy trên ngôn ngữ tiếng Việt [55, 56].

1.2.3. Học hỗn hợp

Học hỗn hợp (Compositionality) được trong *Thị giác Máy tính* được nhắc đến lần đầu trong công trình khoa học "*Part of Recognition*" của Hoffman và Richards [14]. Học hỗn hợp cố gắng phân tích các khái niệm, các thực thể thành các khái niệm và các thuộc tính đơn giản hơn. Trong *Thị giác Máy tính cổ điển*, những mô hình với cấu trúc tượng hình được nghiên cứu một cách rộng rãi [15, 16, 17]. Hiện nay, nhánh nghiên cứu Học hỗn hợp đã trở nên phổ biến trở lại với cộng đồng Deep Learning [13, 18, 19, 20, 21].

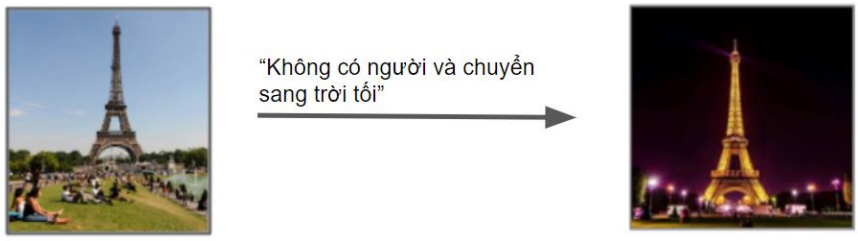
Ví dụ: Để mô tả "con mèo" nhà ta, ta có thể phân rã thành các khái niệm, hoặc thuộc tính đơn giản như sau:

Thuộc tính	Giá trị
Màu sắc	Vàng
Có vuốt?	Có
Độ dài lông	Rất dài

Bảng 1.2. Mô tả đối tượng „mèo“ bằng thuộc tính

Với cách biểu diễn này, khi thực hiện những sự thay đổi nhỏ trên giá trị của các thuộc tính trên của mèo, chúng ta có hoàn toàn có thể tạo ra một thực thể mèo mới hoàn toàn khác mà không cần phải định nghĩa một loài động vật mới.

Trong bài toán mà chúng tôi nghiên cứu, câu mô tả tăng cường là một dạng thể hiện cho sự thay đổi mong muốn trên thuộc tính của bức ảnh tham khảo đầu vào của người dung trên cùng một không gian biểu diễn chung.



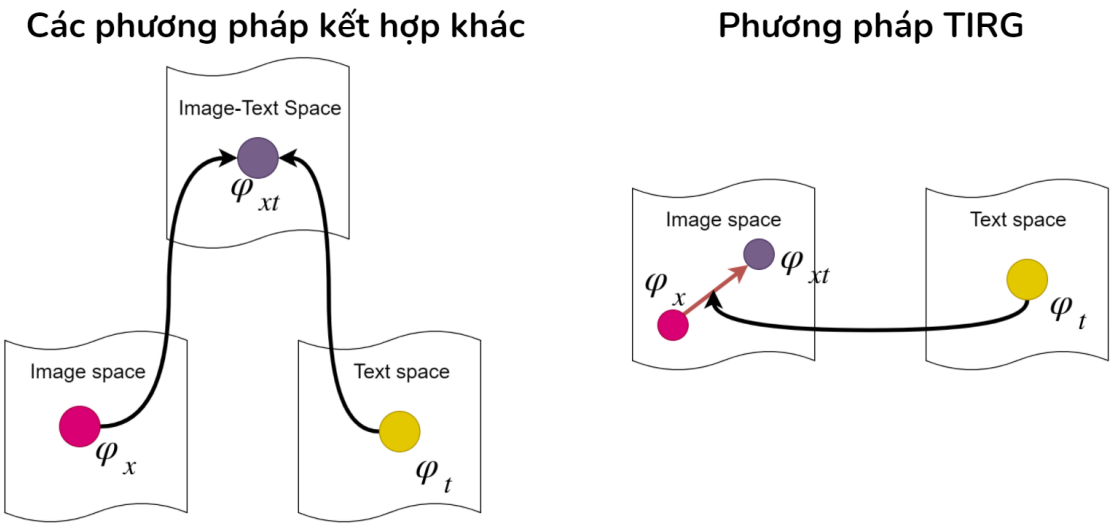
Có người?	Trời sáng?	Có mặt đường?
Có	Có	Có

Có người?	Trời sáng?	Có mặt đường?
Không	**Không**	Có

Ảnh 1.5. Minh họa cho sự thay đổi thuộc tính trong bài toán kết hợp ảnh và câu mô tả tăng cường

1.2.4. Nhận xét

Chúng ta đã có những phương pháp để kết hợp ảnh và văn bản được ứng dụng trong tác vụ **Hỏi đáp trên ảnh**,.. Tuy nhiên ở những bài báo trên, cách kết hợp chủ yếu là xây dựng một không gian biểu diễn "hoàn toàn mới" để dung nạp ảnh và văn bản, cố gắng đưa biểu diễn ảnh về không gian thứ ba, do đó không phù hợp để giải quyết bài toán truy vấn ảnh. Trong đó phương pháp gần với phương pháp chúng tôi nhất là phương pháp **FiLM**, phương pháp này sử dụng ít tham số hơn và chỉ có thể thực hiện một số phép biến đổi đơn giản khi kết hợp ảnh và văn bản, do đó không gian biểu diễn bị hạn chế. Nguyên nhân chính đặc trưng kết hợp này được xây dựng để phục vụ bài toán khác không phải truy vấn, do đó bỏ qua vai trò của biểu diễn ảnh.



Ảnh 1.6. Cách xây dựng đặc trưng biến đổi của các phương pháp khác so với phương pháp **TIRG**. Các phương pháp khác sẽ xây dựng đặc trưng kết hợp bằng cách đưa về không gian chung thứ ba **Image-Text Space**. Còn phương pháp TIRG sử dụng đặc trưng văn bản để dịch chuyển đặc trưng ảnh thành đặc trưng kết hợp trên không gian gốc của ảnh, do đó sẽ phù hợp hơn với bài toán truy vấn.

Chúng ta cũng thấy tình hình nghiên cứu Học máy sôi động của các bài toán Xử lí ngôn ngữ tự nhiên trên tiếng Việt: rất nhiều phương pháp, bài toán và tập dữ liệu đánh giá ra đời trong thời gian gần đây. Đây là một làn gió mới với hi vọng mang những công cụ Học máy để giải quyết các bài toán tiếng Việt của chúng ta, vốn mang tính đặc thù về địa phương cao.

1.3. Mục tiêu nghiên cứu

Sau khi tìm hiểu sơ bộ về bài toán kết hợp ảnh và câu mô tả tăng cường cho truy vấn ảnh, chúng tôi xác định mục tiêu nghiên cứu như sau:

- **(i)** Đầu tiên chúng tôi nghiên cứu và tìm hiểu về các phương pháp cho bài toán kết hợp ảnh và câu mô tả tăng cường cho truy vấn ảnh.
- **(ii)** Chúng tôi nghiên cứu xây dựng mô hình **TIRG** cho bài toán kết hợp ảnh và câu mô tả tăng cường cho truy vấn ảnh, với những cải tiến cụ thể,

để mô hình hoạt động tốt trên dữ liệu tiếng Việt, với những đặc thù về ngôn ngữ rất khác biệt so với tiếng Anh.

- **(iii)** Chúng tôi tiến hành xây dựng tập dữ liệu CSS-VN để phục vụ bài toán trên.

1.4. Đóng góp của khóa luận

- **(i)** Chúng tôi đã nghiên cứu, tìm hiểu và ứng dụng phương pháp **Text-Image Residual Gating (TIRG)** [1] được đề xuất ở hội nghị **CVPR2019** để giải quyết bài toán truy vấn ảnh sử dụng kết hợp ảnh và câu mô tả tăng cường. Qua đó, chúng tôi tái hiện thành công kết quả bài báo trên tập dữ liệu tiếng Việt **CSS-VN** và **MIT-States** với kết quả tương tự bài báo ở trên tập dữ liệu **CSS** tiếng Anh gốc, đồng thời rút trích được những bài học và góc nhìn rất thú vị về mô hình trên.
- **(ii)** Chúng tôi ***cải tiến*** thành công mô hình **TIRG** bằng cách thay thế bộ biểu diễn từ (Word Embedder) của **TIRG** từ một lớp **Embedding** thành một *mô hình ngôn ngữ* được huấn luyện trên dữ liệu khổng lồ là **PhoBERT**, cho phép mô hình *thích ứng* với những từ *nằm bên ngoài từ điển* của bộ dữ liệu huấn luyện. Đồng thời, để mô hình hoạt động hiệu quả, chúng tôi còn sử dụng bộ công cụ tách từ **RDRSegmenter** để đối phó với hiện tượng nhặp nhằng khoảng trắng ở tiếng Việt.
- **(iii)** Chúng tôi xây dựng thành công bộ dữ liệu CSS-VN tiếng Việt sử dụng công cụ **URBANS** [5] như một công cụ chính yếu và duy nhất trong toàn bộ quá trình dịch bộ dữ liệu. Bộ dữ liệu này sau đó được sử dụng cho nghiên cứu của chúng tôi trong việc tìm hiểu cũng như đánh giá trên phương pháp mà chúng tôi chọn. Kết quả khi áp dụng mô hình trên tập dữ liệu **CSS-VN** tốt do những lợi thế đặc thù về ngôn ngữ của tiếng Việt khi được huấn luyện trên mô hình Học máy.
- **(iv)** Chúng tôi xây dựng thành công một bộ công cụ dịch dựa trên tập luật **URBANS** [5] và sử dụng nó để dịch bộ dữ liệu **CSS** được đề xuất trong bài

báo [1]. Bộ công cụ này là một mã nguồn mở và được đăng tải trên *pypi* để tất cả mọi người có thể tải xuống và sử dụng. Ngoài ra chúng tôi cũng xây dựng một kịch bản kiểm thử hoàn chỉnh cho bộ công cụ này để tránh những sai sót trong quá trình phát triển mã nguồn mở. (*https://github.com/pyurbans/urbans*)

1.5. Cấu trúc khóa luận tốt nghiệp

Phần còn lại của khóa luận tốt nghiệp sẽ được chúng tôi tổ chức như sau:

Chương 2 chúng tôi sẽ giới thiệu một kiến thức nền tảng phục vụ cho việc giải quyết bài toán Truy vấn ảnh dựa trên ảnh và câu truy vấn tăng cường

Chương 3 chúng tôi sẽ chia sẻ về công cụ dịch dựa trên cây cú pháp và quy trình chúng tôi xây dựng ra tập dữ liệu tiếng Việt CSS-VN

Chương 4 chúng tôi sẽ tập trung chia sẻ về hướng tiếp cận cho bài toán truy vấn ảnh dựa trên ảnh và câu mô tả

Chương 5 chúng tôi sẽ trình bày các kết quả thí nghiệm, đồng thời chia sẻ những kết luận và góc nhìn của chúng tôi về các thí nghiệm trên

Chương 6 chúng tôi sẽ đưa ra kết luận ngắn gọn về kết quả nghiên cứu khóa luận và hướng nghiên cứu tiềm năng cho bài toán của chúng tôi

Chương 2. **KIẾN THỨC NỀN TẢNG**

Trong chương này, chúng tôi trình bày một kiến thức nền tảng về truy vấn thông tin và mạng nơ-ron. Nội dung chương này nhắc đến tổng quan bài toán truy vấn thông tin, các kiến trúc để biểu diễn ảnh và văn bản dựa trên mạng học sâu và cách huấn luyện, là nền tảng cốt lõi cho phương pháp mà chúng tôi sử dụng cho bài toán Truy vấn ảnh sử dụng kết hợp ảnh và câu mô tả tăng cường.

2.1. Tổng quan về bài toán truy vấn thông tin

Từ xa xưa, loài người cổ đại đã phải trang bị rất nhiều kĩ năng để phục vụ cho việc sinh tồn: Săn bắn, hái lượm, leo trèo,... Mà trong đó, tìm kiếm là một trong những kĩ năng sống còn của con người. Theo dòng thời gian, với sự xuất hiện của chữ viết và sách, việc lưu trữ và tìm kiếm lại trở thành một kĩ năng sinh tồn.

2.1.1. Truy vấn thông tin là gì?

Thuật ngữ *Truy vấn thông tin* có thể mang nghĩa rất rộng. Khi đi mua hàng, bạn lấy thẻ tín dụng từ trong ví ra để có thể nhập mã thẻ thanh toán, đó cũng là một dạng của truy vấn thông tin.

Tuy nhiên, ở khía cạnh học thuật, Truy vấn Thông tin được định nghĩa là:

> Truy vấn thông tin là hoạt động tìm kiếm tài liệu có bản chất phi cấu trúc như văn bản, hình ảnh, video,.. sao cho phù hợp với một nhu cầu thông tin nào đó, từ một tập hợp dữ liệu lớn. [24]

Đầu vào của một hệ thống truy vấn văn bản tiêu biểu:

- Một bộ ngữ liệu các tài liệu văn bản
- Một câu truy vấn của người dùng dưới dạng văn bản

Đầu ra:

- Một tập các văn bản được cho là phù hợp với truy vấn đầu vào, được sắp xếp theo độ phù hợp giảm dần

2.1.2. Thế nào là một kết quả trả về phù hợp ?

Tính phù hợp của một kết quả trả về là một đánh giá mang tính chủ quan (và có thể) bao gồm:

- Đúng chủ đề
- Đúng thời điểm
- Đáng tin cậy
- Thỏa mãn mục tiêu và ý định của người tham gia tìm kiếm về nhu cầu thông tin

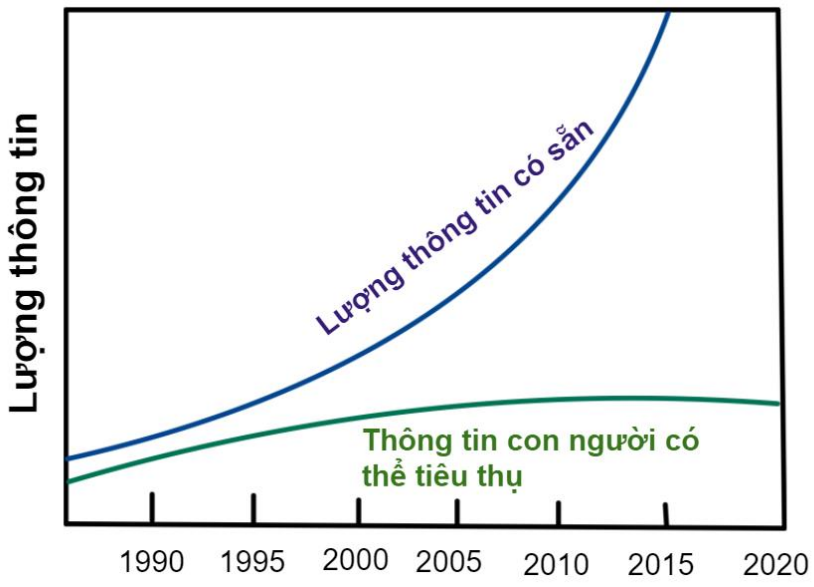
Ảnh 2.1. Hình minh họa về tính phù hợp

2.1.3. Động lực của Truy vấn Thông tin

2.1.3.1. Quả tải thông tin

Với sự phát triển vũ bão của dữ liệu, *Truy vấn thông tin* được sinh ra để giải quyết vấn đề *quá tải thông tin.*

Quá tải thông tin là sự khó khăn trong việc tiếp thu và đưa ra quyết định hiệu quả với một vấn đề khi tồn tại quá nhiều thông tin về vấn đề đó.

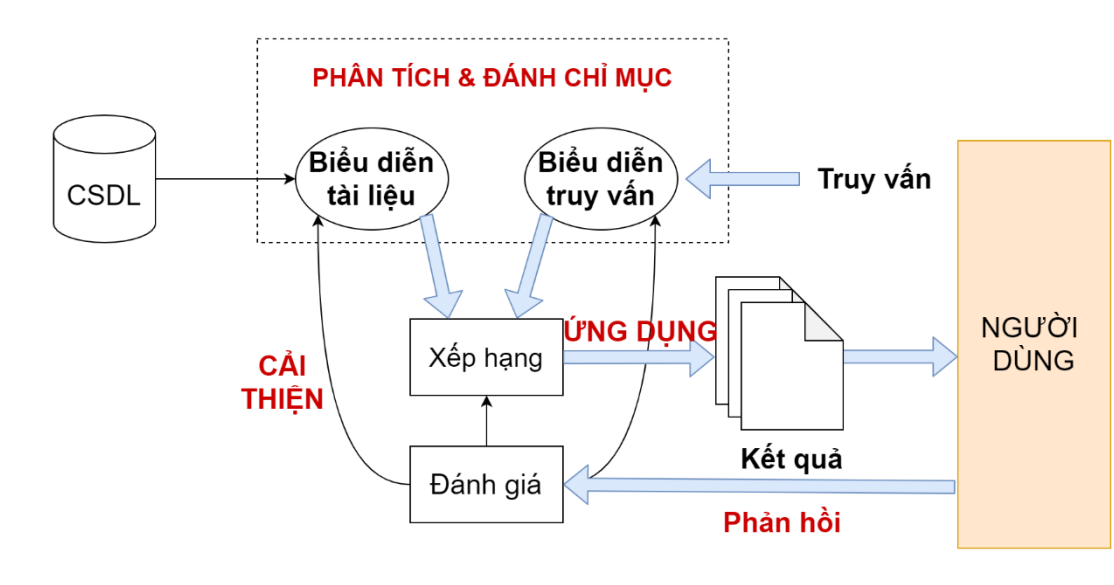


Ảnh 2.2. Dữ liệu càng ngày càng vượt xa giới hạn con người có thể tiếp thu

2.1.3.2. Làm việc với dữ liệu phi cấu trúc

Mục tiêu của *Truy vấn thông tin* còn là để giúp ta làm việc với *dữ liệu phi cấu trúc* do tính chất phức tạp của chúng. Để làm việc với dữ liệu có cấu trúc chúng ta đã có những công cụ rất mạnh như những cơ sở dữ liệu quan hệ và truy vấn trên cơ sở dữ liệu này. Tuy nhiên, đặc điểm của dữ liệu phi cấu trúc rất đặc biệt:

- Tồn tại ở nhiều dạng khác nhau: email, hình ảnh, video, âm thanh,..
- 85% dữ liệu của một doanh nghiệp tồn tại ở dạng phi cấu trúc, theo Merril Lynch
- Ngữ nghĩa không rõ ràng Mô hình thực hiện truy vấn điển hình



Ảnh 2.3. Mô hình truy vấn tiêu biểu

Trong một hệ thống *Truy vấn Thông tin* điển hình, bộ ngữ liệu sẽ được biểu diễn và lưu trữ trước tại trong cơ sở dữ liệu. Sau đó, với mỗi truy vấn của người dùng, hệ thống sẽ thực hiện việc biểu diễn câu truy vấn đó và sử dụng phép biểu diễn đó để đi so sánh với các phép biểu diễn hiện có ở trong cơ sở dữ liệu bằng một độ đo khoảng cách nhất định. Những tài liệu với biểu diễn gần và tương đồng nhất với câu truy vấn sẽ được trả về ở giao diện người dùng dưới dạng một danh sách các tài liệu, được sắp xếp theo độ phù hợp giảm dần.

Sau đó, người dùng hoặc người triển khai hệ thống tìm kiếm sẽ thực hiện đánh giá lại hệ thống tìm kiếm dựa trên mức độ phù hợp của kết quả trả về bằng các phương pháp đánh giá cụ thể, từ đó đưa ra các hướng phát triển để cải thiện hệ thống truy vấn một cách phù hợp.

2.1.4. Đánh giá hệ thống truy vấn thông tin

[24] Để đánh giá một hệ thống truy vấn một cách theo một cách hiệu quả và tiêu chuẩn, chúng ta cần có một tập **Tiêu chuẩn vàng** (hay **Gold Standard**) gồm các thành phần sau:

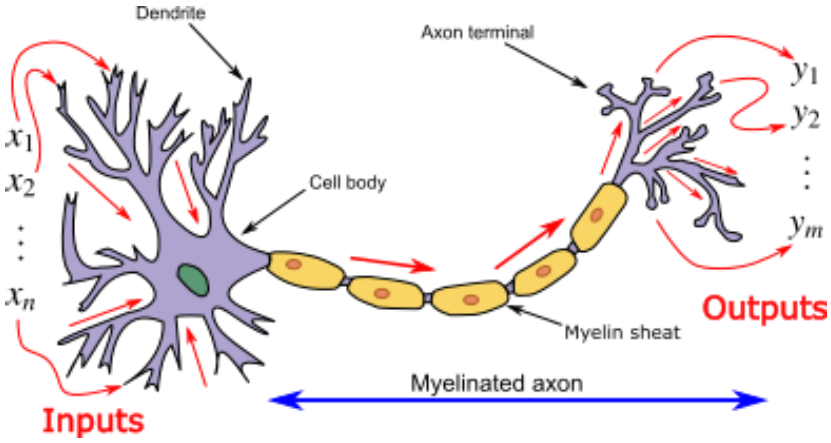
1. Một cơ sở dữ liệu (có thể là ảnh, văn bản hay âm thanh,..)
2. Một tập các nhu cầu thông tin, được biểu diễn bằng câu truy vấn
3. Một tập đánh giá về tính phù hợp cho mỗi cặp truy vấn - dữ liệu mục tiêu.

Trong đó:

- **Tiêu chuẩn vàng**: là tập dữ liệu dùng để đánh giá một hệ thống tìm kiếm. Việc đánh giá một hệ thống tìm kiếm xoay quanh việc đánh giá tính phù hợp của kết quả trả về trên hệ thống tìm kiếm đó. Cho trước một nhu cầu thông tin, một tài liệu được cho trong tập đánh giá được dán nhãn là phù hợp hay không phù hợp với câu truy vấn đầu vào bất kì.
- **Tính phù hợp:** được đánh giá dựa trên nhu cầu thông tin, không phải câu truy vấn. Giả sử người dùng nhập câu truy vấn là "mắt biếc", người dùng có thể đang tìm kiếm những tấm ảnh về đôi mắt biếc, đang muốn tìm hiểu xem một đôi mắt biếc trông như thế nào. Người dùng cũng có thể đang tìm kiếm tác phẩm Mắt Biếc của tác giả Nguyễn Nhật Ánh, để có thể mua về đọc.

2.2. Tổng quan về mạng nơ-ron nhân tạo

Mạng nơ-ron nhân tạo được giới thiệu lần đầu tiên vào năm 1985 [9], lấy ý tưởng từ mạng nơ-ron sinh học của người. Ở mạng nơ-ron ở người, các tín hiệu xung thần kinh được dẫn truyền qua các đơn vị thần kinh cơ bản nhất, là nơ-ron và các axon.

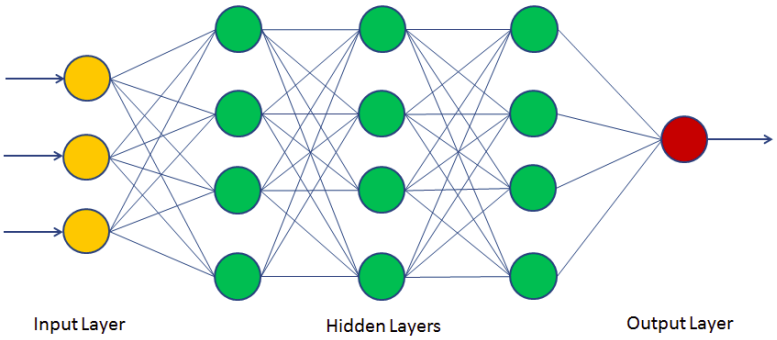


Ảnh 2.4. Mạng nơ-ron ở người

Mục tiêu của mạng nơ-ron nhân tạo là tìm cách xấp xỉ một hàm bằng cách sử dụng tổ hợp các hàm phi tuyến đơn giản.

Gần đây với sự phát triển của các kiến trúc máy tính phù hợp cho việc tính toán song song, điển hình là card đồ họa (hay GPU), mạng nơ-ron đã phát triển hơn bao giờ hết. Khởi đầu cho phong trào này với mô hình mạng nơ-ron tích chập AlexNet (2012) với chiến thắng tại giải thưởng ImageNet Large Scale Visual Recognition Challenge (ILSVRC2012) vào tháng 9 năm 2012.

2.2.1. Mạng lan truyền thuận đa lớp



Ảnh 2.5. Mạng lan truyền thuận đa lớp

Một mạng nơ-ron điển hình được biểu diễn bằng các lớp tính toán bao gồm các phép biến đổi phi tuyến mang tính tuần tự, biến đổi một tensor này sang tensor

khác thông qua các lớp liền kề nhau. Trong ví dụ sau, chúng tôi xin giới thiệu một mạng lan truyền thuận đơn giản, bao gồm hai lớp.

Trong đó, phương pháp tối ưu phổ biến nhất được dùng để tối ưu mạng lan truyền thuận đa tầng là phương pháp **Gradient Descent**. Chúng tôi sẽ đề cập tới nó ở phần sau.

2.2.1.1. Lớp ẩn

Lớp ẩn là một trong đơn vị thành phần cấu thành của mạng nơ-ron lan truyền thuận đa tầng. Đầu ra của một lớp ẩn này sẽ là đầu vào của lớp ẩn kia. Mỗi lớp ẩn được cấu thành bởi một loạt các đơn vị ẩn (hidden unit). Mục tiêu của lớp ẩn là xây dựng các phép biểu diễn, các đặc trưng bằng cách kết hợp các đặc trưng của lớp trước đó. Việc thiết kế lớp ẩn được dựa trên kinh nghiệm và trực giác của người thiết kế lớp ẩn. Một số lớp ẩn nổi bật có thể kể đến như là residual block trong ResNet50 hay Inception layer của GoogleNet.

2.2.1.2. Hàm kích hoạt

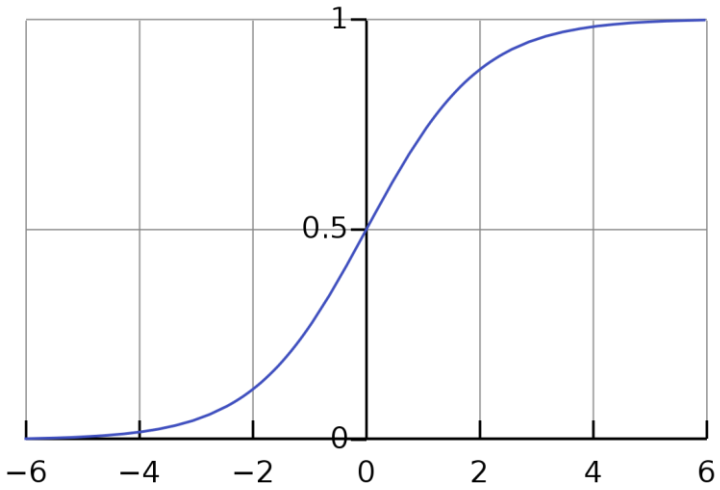
Mạng nơ-ron ngoài ngoài được đặc trưng bởi các toán tử lan truyền thuận như toán tử tích chập hay biến đổi affine, nó còn được đặc trưng bởi các hàm kích hoạt. Hàm kích hoạt sẽ quyết định kết quả của các toán tử tuyến tính sẽ tiếp tục được biến đổi như thế nào tại các nút trong mạng nơ-ron. Ở đây, chúng tôi xin được kể tên một số loại hàm kích hoạt cơ bản thông dụng, thường dùng trong những mạng nơ-ron tiêu chuẩn.

Thông thường, một lớp ẩn sẽ bao gồm một toán tử phép biến đổi tuyến tính và một hàm kích hoạt.

Hàm kích hoạt Sigmoid

Hàm kích hoạt Sigmoid là hàm kích hoạt được sử dụng nhiều trong bài toán phân lớp nhị phân. Hàm sigmoid lấy đầu vào là một giá trị bất kì và ánh xạ thành một giá trị trong khoảng [0,1]

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (1)$$



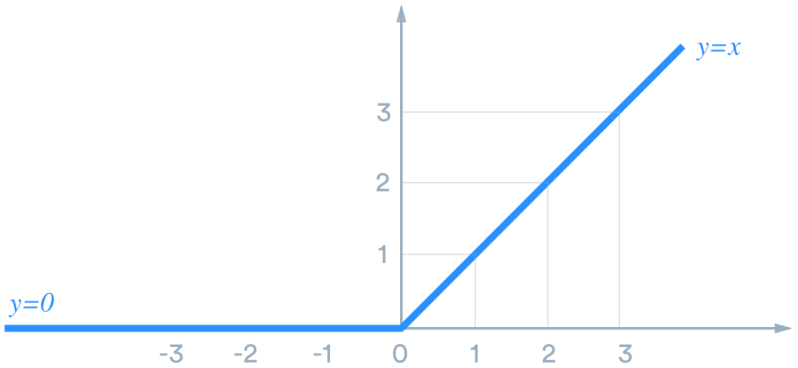
Ảnh 2.6. Hàm kích hoạt Sigmoid

Tuy nhiên hàm sigmoid không được sử dụng để làm hàm kích hoạt ở giữa mạng nơ-ron do xảy ra hiện tượng tan biến gradient khi thực hiện lan truyền ngược với đầu vào x khi $x \to \pm\infty$.

Hàm kích hoạt ReLU

Hàm ReLU (Rectified Linear Unit) là một hàm kích hoạt được sử dụng rất phổ biến khi xây dựng mạng nơ-ron đa lớp. Khắc phục nhược điểm tiêu biến gradient (Gradient Vanishing) của hàm sigmoid.

$$y = R(z) = \max(0, z) \quad (2)$$



Ảnh 2.7. Hàm kích hoạt ReLU

Đối với hàm kích hoạt ReLU, đạo hàm của y theo biến x luôn cho ra giá trị 1 với $x > 0$.

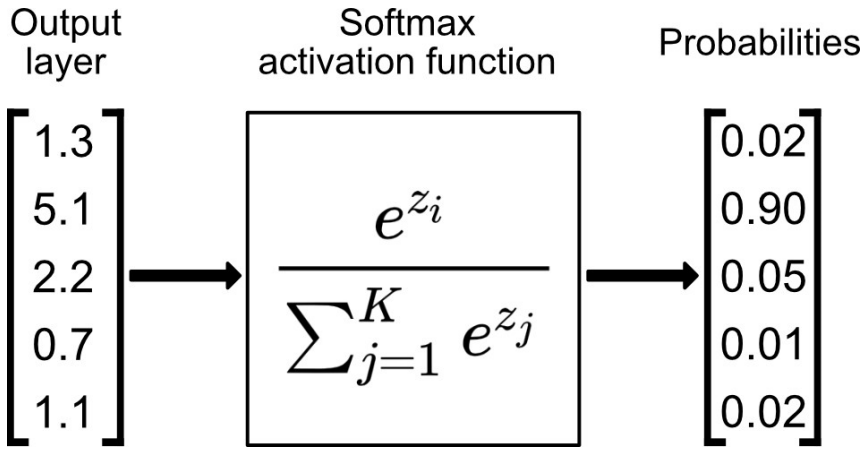
Hàm kích hoạt Softmax

Hàm kích hoạt softmax là một hàm lấy đầu vào là một vector logits z, ánh xạ thành một vector chứa một *phân bố xác suất* (có tổng là 1).

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad (3)$$

Trong đó:

- z_i là phần tử thứ i của vector z
- K là số phần tử của vector z



Ảnh 2.8. Hàm kích hoạt Softmax

Hàm softmax thường được dùng trong các bài toán phân loại đa lớp dựa trên mạng nơ-ron do đặc thù đầu ra là một *phân phối xác suất*. Trong bài toán phân loại đa lớp, cho trước mẩu dữ liệu x ta muốn ước lượng xác suất của x rơi vào lớp thứ i là bao nhiêu. Hàm softmax thỏa mãn điều kiện này vì đầu ra của hàm

softmax luôn là một phân phối xác suất, các phần tử luôn dương và có tổng bằng một.

Với $K = 2$, hàm softmax tương đương với hàm sigmoid.

2.2.2. Hàm mất mát

Hàm mất mát là một trong những thành phần cấu thành quan trọng trong khi huấn luyện mạng nơ-ron nhân tạo, hàm mất mát cho biết độ lỗi của mô hình với kết quả lý tưởng là bao xa. Dưới đây chúng tôi xin giới thiệu một số hàm mất mát cơ bản cho bài toán

2.2.2.1. Mean Square Error

Trong bài toán hồi quy tiêu chuẩn, ta xây dựng một mô hình tham số hóa f: $(\theta, X) \rightarrow \hat{y}$ với tập dữ liệu huấn luyện gồm đầu vào X và nhãn là y. Ta muốn đo chất lượng của dự đoán $\hat{y}$. Thông thường Mean Square Error (MSE) là một hàm mất mát được sử dụng dụng để huấn luyện các mô hình hồi quy:

$$MSE(y, \hat{y}) = \frac{1}{M} \sum_{i=0}^{M} (\hat{y} - y)^2 \qquad (4)$$

Có thể thấy một cách trực quan, giá trị MSE nhỏ khi khoảng cách giữa giá trị dự đoán đầu ra và nhãn càng gần nhau và ngược lại, sẽ lớn khi giá trị dự đoán cách xa nhãn của điểm dữ liệu đó.

2.2.2.2. Categorical Cross-Entropy

Categorical Cross-Entropy là hàm mất mát được sử dụng nhiều trong bài toán phân loại đa lớp. Categorical Cross-Entropy nhận vào là hai phân phối xác suất y và $\hat{y}$. Categorical Cross-entropy là hàm đối log hợp lý (negative log-likelihood)

$$CE(y, \hat{y}) = -\sum_{i} y_i \log \hat{y}_i \quad (5)$$

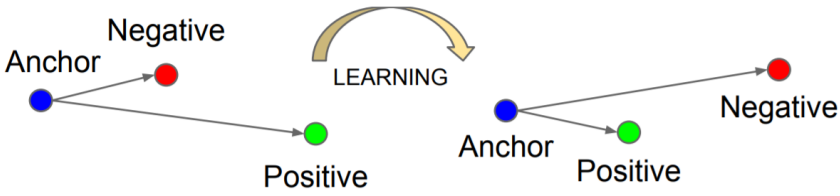
Cross-entropy sẽ cho ra giá trị nhỏ nếu phân phối xác suất y và $\hat{y}$ càng khớp nhau và ngược lại.

2.2.2.3. Triplet loss

Triplet loss là một hàm mất mát được đề xuất trong bài báo [2]. Đây là một hàm mất mát được đề xuất để phục vụ cho bài toán **học biểu diễn** (*representation learning*) một cách hiệu quả. So với các nghiên cứu trước đó, vốn đơn thuần là học phép biểu diễn như là một phần của bài toán phân lớp, Triplet Loss nhắm vào việc tìm ra một phép biểu diễn mang tính ý nghĩa cao bằng việc đặt thêm các *ràng buộc về khoảng cách* cho các thực thể vectơ trong không gian biểu diễn. **Triplet Loss** và được sử dụng trong mạng **Triplet Network** và được chứng minh là tốt hơn so với người tiền nhiệm là **Siamese Network** [3] vốn dĩ dựa trên việc học phép biểu diễn trên một bài toán phân lớp.

Mục tiêu của **Triplet Loss** là tìm một phép biểu diễn sao cho những đối tượng (ví dụ ảnh, văn bản,..) có yếu tố ngữ nghĩa tương đồng nhau thì sẽ nằm *tiệm cận* nhau trên không gian biểu diễn. Tương tự, những đối tượng có yếu tố ngữ nghĩa *không liên quan* hoặc *tương phản nhau* sẽ nằm xa nhau trên không gian biểu diễn đó.

Trong đó, với mỗi mẫu huấn luyện, ta sẽ có **mẫu cột mốc** (*anchor*) tương ứng với **mẫu phù hợp** (*positive*) và **mẫu tương phản** (negative).



Ảnh 2.9. Minh họa Triplet Loss. Mục tiêu

$$Loss = \sum_{i=1}^{N} \left[\left\| f_i^a - f_i^p \right\|_2^2 - \| f_i^a f - f_i^n \|_2^2 + \alpha \right]_+ \quad (6)$$

Trong đó:

- f^a là biểu diễn của mẫu cột mốc
- f^p là biểu diễn của mẫu phù hợp
- f^n là biểu diễn của mẫu tương phản
- α là **khoảng cách biên** (*margin*), dùng để khuếch đại khoảng cách từ mẫu cột mốc tới mẫu phù hợp và khuếch đại khoảng cách từ mẫu cột mốc tới mẫu tương phản bằng 1 giá trị biên.

2.2.3. Huấn luyện mạng nơron

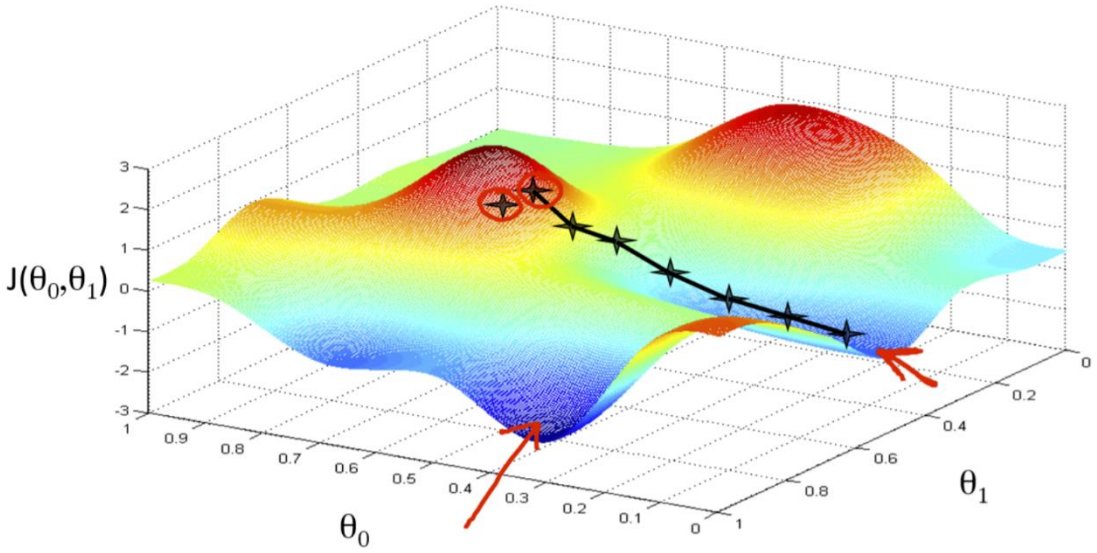
2.2.3.1. Gradient Descent

Gradient Descent là một phương pháp tối ưu tham số mạng nơ-ron bằng việc sử dụng công cụ đạo hàm trong Toán Giải tích. Gradient Descent cho phép chúng ta *cập nhật liên tục* bộ trọng số của mạng nơ-ron đi *ngược chiều đạo hàm* bộ trọng số đối với **hàm độ lỗi**, đi từng bước nhỏ cho đến khi tới được với điểm cực tiểu trên bề mặt hàm lỗi.

Giả sử ta có một hàm số $f: x, \theta \rightarrow y$ trong đó ta muốn tìm bộ tham số θ sao cho cực tiểu y.

Đầu tiên ta tính đạo hàm của θ đối với y. Đạo hàm này sẽ cho biết độ dốc của y tại điểm θ. Sau đó ta cập nhật theta với đi ngược chiều với đạo hàm này bằng phép toán trừ, được tỉ lệ bởi một số siêu tham số α, gọi là **tốc độ học** (learning rate). Việc lựa chọn con số α sẽ tùy vào chiến lược huấn luyện của người kĩ sư Học Máy.

$$\theta := \theta - \alpha \frac{\partial y}{\partial \theta} \tag{7}$$

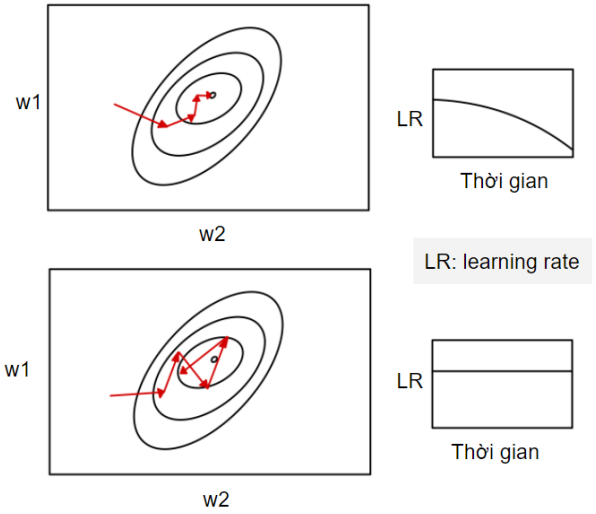


Ảnh 2.10. Minh họa của Gradient Descent [36]

Ban đầu chúng ta sẽ khởi tạo bộ trọng số của mạng nơ-ron, sau đó cập nhật bộ trọng số này ngược chiều đạo hàm đối với hàm loss tỉ lệ với siêu tham số α gọi là **tốc độ học** (hay learning rate).

2.2.3.2. Learning rate decay

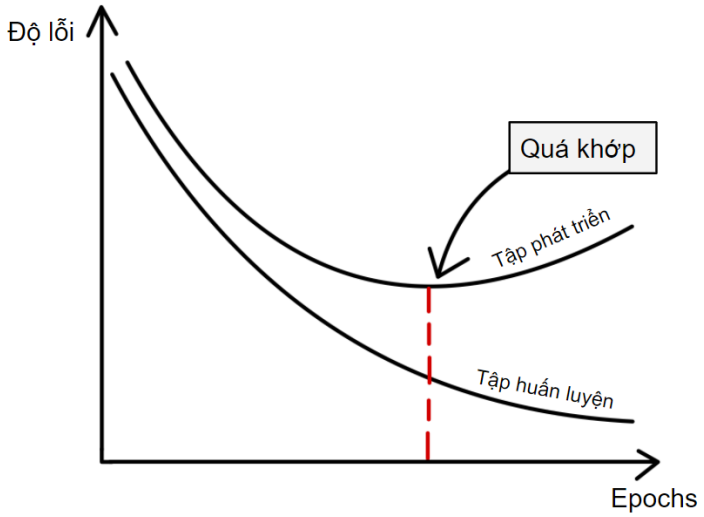
Learning rate decay là một kĩ thuật thường được sử dụng trong việc huấn luyện mạng nơ-ron bằng **Stochastic Gradient Descent**. Ý tưởng chính của Learning rate decay là cho phép mô hình tuần tự giảm **tốc độ học** vào các epochs về sau để mô hình có thể nhanh chóng hội tụ thay vì chỉ dao động xung quanh điểm cực tiểu.



Ảnh 2.11. Minh họa cho kĩ thuật learning rate decay.

2.2.3.3. Early Stopping

Early Stopping là một kĩ thuật được sử dụng trong khi huấn luyện một mô hình *Học Máy*. Bằng Early Stopping, bạn không cần phải khai báo số epochs huấn luyện cụ thể mà có thể huấn luyện vô hạn cho đến khi mô hình không còn cải thiện nữa.



Ảnh 2.12. Ảnh minh họa về Early Stopping

Khi huấn luyện một *mô hình Học Máy*, ta mong muốn rằng sau khi huấn luyện, mô hình chúng ta có thể hoạt động tốt trên các dữ liệu mới, mà ở đây chúng ta

mô phỏng dữ liệu mới này thành một tập dữ liệu gọi là tập phát triển. Ta huấn luyện mô hình với tập huấn luyện và quan sát độ đánh giá của nó trên tập phát triển, khi mô hình không còn cải thiện độ đo đánh giá của nó trên tập huấn luyện nữa, chúng ta dừng quá trình huấn luyện lại. Đây gọi là kĩ thuật **Early Stopping**, được nhắc đến trong sách **Pattern Recognition and Machine Learning của Bishop** [23].

Quá trình huấn luyện được dừng lại và mô hình này được sử dụng và cho rằng là có tính khái quát cao. Đây cũng được coi là một trong những phương pháp **Regularization** cho mạng nơ-ron. Nếu **Weight Decay** là một phương pháp regularization tường minh thì **Early Stopping** là một phương pháp phi tường minh.

2.3. Mô hình mạng nơ-ron tích chập cho biểu diễn ảnh kỹ thuật số

Mạng nơ-ron tích chập là một mạng nơ-ron được thiết kế để xử lý các dạng *dữ liệu dạng lưới*, mang tính không gian. Mạng nơ-ron tích chập chủ yếu sử dụng phép toán tích chập là thành phần cấu thành chính của mạng. Ứng dụng chủ yếu của nơ-ron tích chập là trong xử lý dữ liệu ảnh hay chuỗi thời gian.

2.3.1. Cơ sở lý thuyết

Công thức của tích chập ở thời điểm t của 2 tín hiệu f và g là:

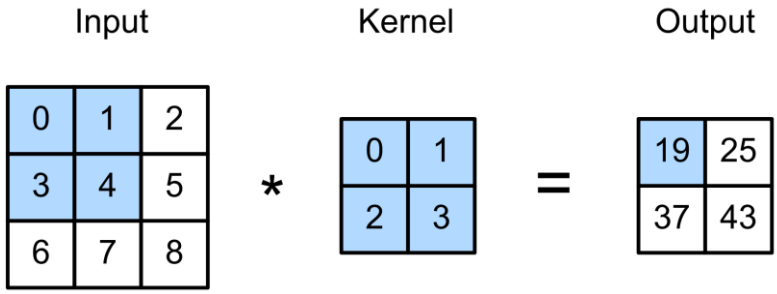
$$(f * g)(t) = \int_{-\infty}^{+\infty} f(x)(t-x)dx \tag{8}$$

Trong đó t là thời điểm nơi mà phép tích chập được thực hiện từ tín hiệu f được lên trên tín hiệu g

Trong bài toán xử lý ảnh sử dụng mạng nơ-ron tích chập, phép tích chập được thực hiện 2 chiều, trên biến chạy thuộc miền rời rạc.

$$f[x,y] * g[x,y] = \sum_{n_1=-\infty}^{+\infty} \sum_{n_2=-\infty}^{+\infty} f[n_1,n_2].\, g[x-n_1, y-n_2] \qquad (9)$$

Thông thường, khi được cài đặt trên các thư viện lập trình, mạng nơ-ron tích chập sử dụng phép toán **tương quan chéo** (cross-correlation) thay cho phép tích chập do dễ cài đặt và mang lại kết quả tương đồng.



Ảnh 2.13. Thực hiện phép tích chập với đầu vào trên một bộ lọc cho trước

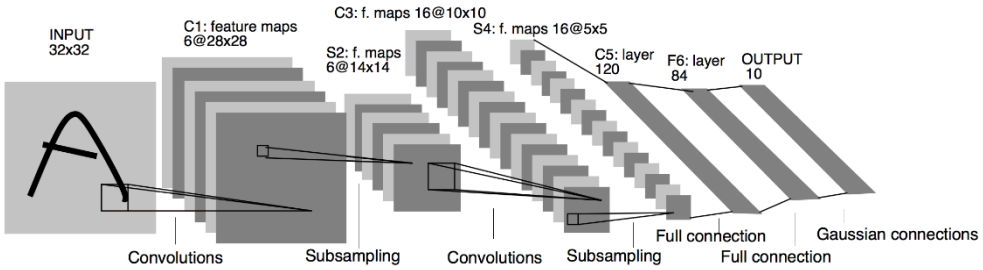
Một mạng nơ-ron tích chập thông thường sẽ có ba thành phần cấu thành chính:

- Lớp **tích chập** *(Convolution):* cho phép rút trích thông tin cục bộ của bức ảnh thông qua phép tích chập trên Tensor đầu vào thông qua các bộ lọc. Các tham số của các bộ lọc này sẽ được học trong quá trình huấn luyện.
- Lớp **chiết xuất** *(Pooling)*: dùng để giảm tham số của mạng nơ-ron tích chập bằng cách giảm kích thước của bản đồ đặc trưng của từ lớp tích chập trước đó. Hai phương pháp chiết xuất phổ biến là chiết xuất cực đại (Max Pooling) và chiết xuất trung bình (Average Pooling).
- Lớp **kết nối đầy đủ** *(Fully connected)*: thường được dùng ở các lớp cuối của mạng nơ-ron tích chập để trích xuất các đặc trưng toàn cục để phục vụ cho bài toán khác, ví dụ như bài toán phân lớp.

2.3.2. Quá trình phát triển của mạng nơ-ron tích chập

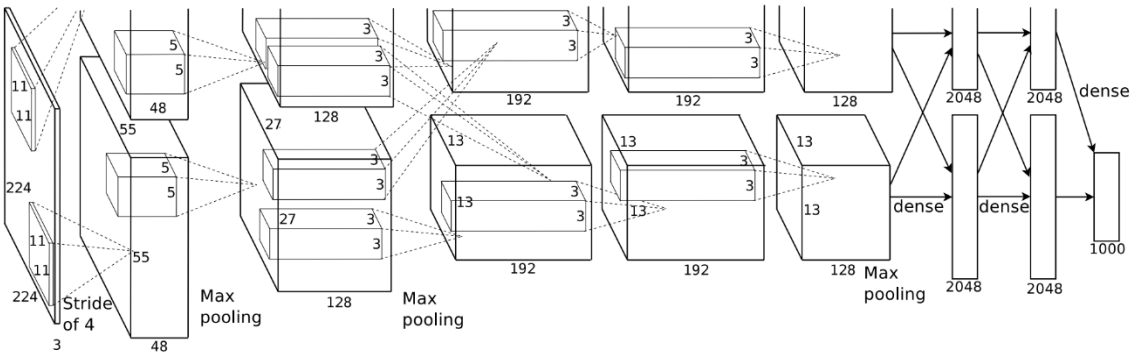
Mạng nơ-ron tích chập tuy xuất hiện chưa lâu nhưng đã có những bước tiến và thành tựu đáng nể phục, chúng ta sẽ cùng nhìn lại quá trình phát triển của một số kiến trúc mạng nơ-ron tích chập nổi bật từ khi được giới thiệu cho đến nay.

Mạng nơ-ron tích chập lần đầu tiên được giới thiệu bởi Yann Lecun với bài toán phân lớp chữ viết bằng cách huấn luyện mạng nơ-ron trên tập dữ liệu **MNIST**, gọi là mạng **LeNet-5** [26]. Mạng **LeNet-5** được cấu thành bởi hai lớp tích chập, hai lớp chiết xuất và ba lớp kết nối đầy đủ. Trong đó số lượng lớp có chứa tham số là 5, bao gồm các lớp tích chập và lớp kết nối đầy đủ.



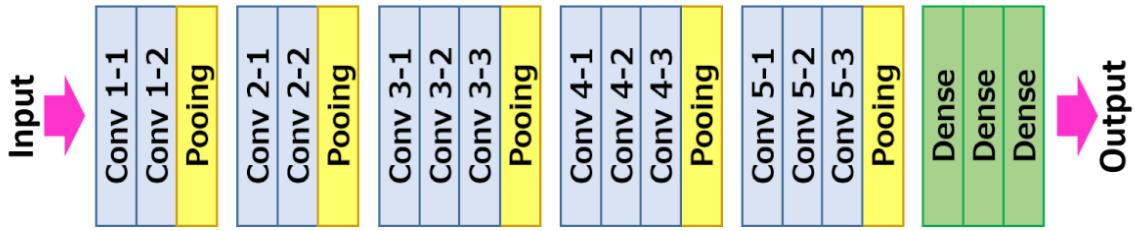
Ảnh 2.14. Mạng nơ-ron tích chập LeNet-5

Năm 2012, Alex Krizhevsky và các cộng sự đã chiến thắng tại *ILSVRC-2012* với mô hình **AlexNet** [38] với 60 triệu tham số và 650,000 nơ-ron. Mạng **AlexNet** bao gồm năm lớp tích chập, trong đó một vài lớp được theo sau bởi lớp chiết xuất cực đại, và kết thúc bằng lớp kết nối đầy đủ với kích hoạt softmax gồm 1000 nút. Để cải thiện hiện tượng *quá khớp* (Overfitting) khi huấn luyện mạng, tác giả còn áp dụng một phương pháp *chính quy hóa* (Regularization) gọi là Dropout, cho phép vô hiệu hóa việc cập nhật trọng số một cách ngẫu nhiên của một số kết nối khi thực hiện lan truyền ngược trên mạng.



Ảnh 2.15. Mạng AlexNet

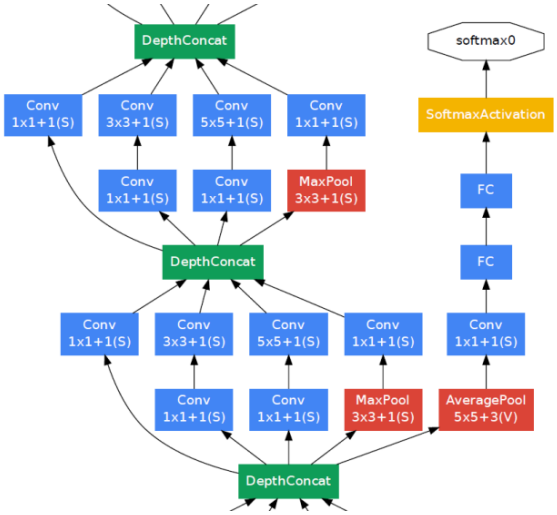
Năm 2014, Karen Simonyan và Andrew Zisserman đã giới thiệu kiến mạng tích chập là **VGG** [39]. Mạng **VGG** là được ra đời nhằm thử thách mức độ sâu của mạng nơ-ron bằng cách tăng số lớp lên đến 16-19 lớp, đồng thời sử dụng bộ lọc *3x3*, nhỏ hơn các bộ lọc của các mạng tiền nhiệm rất nhiều.



Ảnh 2.16. Mạng VGG-16

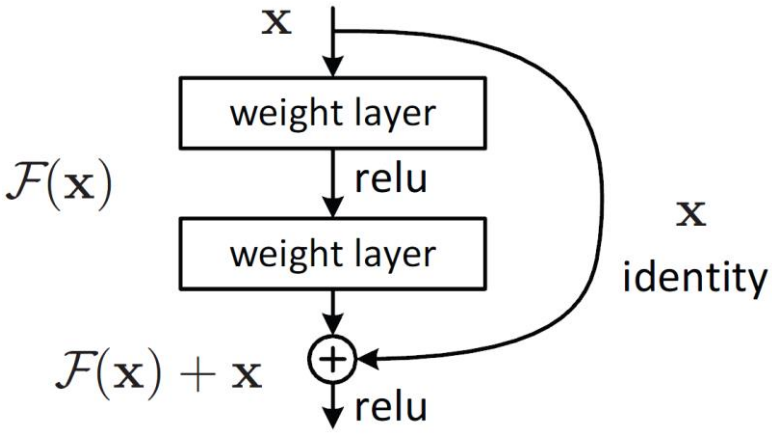
Độ sâu của mạng ngày càng tăng lên cũng đồng nghĩa với việc việc huấn luyện ngày càng trở nên khó khăn. Mạng càng sâu sẽ xảy ra hiện tượng *tiêu biến gradient* khi thực hiện cập nhật trọng số khi lan truyền ngược thông qua quá nhiều lớp. Việc thêm quá nhiều lớp không những không tăng độ chính xác của mạng, mà còn có thể khiến chúng tệ hơn. [27]

GoogLeNet [40](hay **Inception**) được sinh ra nhằm giải quyết hiện tượng tiêu biến gradient. **GoogLeNet** triệt tiêu hiện tượng tiêu biến gradient bằng cách sử dụng xây dựng cái nút phân lớp ở các điểm chính giữa mạng, điều này đảm bảo đặc trưng rút ra từ những lớp đầu tiên có thể đủ tốt để thực hiện việc phân lớp, đồng thời giảm sự thất thoát gradient khi lan truyền ngược về các lớp đầu tiên của mạng.



Ảnh 2.17. Một phần được cắt xén trong mạng **GoogLeNet**

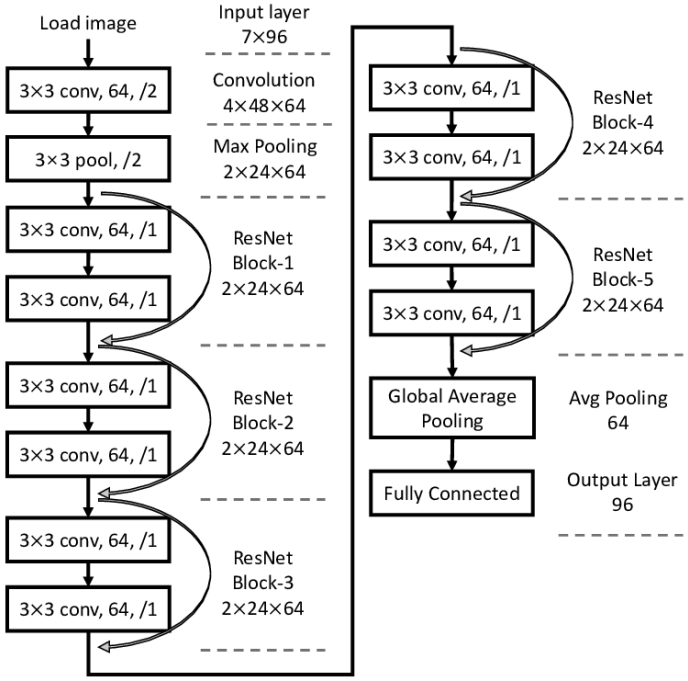
Năm 2015, **Kaiming He** cùng các cộng sự ở **Microsoft Research** đã giới thiệu ra kiến trúc mạng **ResNet** [41] với xây dựng những mạng rất sâu nhưng lại không đánh đổi về độ chính xác. Thành phần cấu thành quan trọng nhất của mạng **ResNet** là **khối nối tắt** (identity block),



Ảnh 2.18. Khối nối tắt trong mạng ResNet

Khối nối tắt cho phép giá trị của những lớp trước được thêm vào những lớp sau để có thể lan truyền ngược trực tiếp mà không phải thông qua các lớp trung gian, hạn chế sự tiêu biến gradient. Với đầu vào là khối tensor X, với phép biến đổi $\boldsymbol{H}(\boldsymbol{X}) = \boldsymbol{F}(\boldsymbol{X}) + \boldsymbol{X}$. Cho dù hàm F($X$) không học được ra gì, hay còn gọi là

lớp chết, thì vẫn có thông tin từ khối tensor X trước đó dẫn lên, hay $H(X) = X$ khi $F(X) = 0$.



Ảnh 2.19. Mạng ResNet-12

Có thể thấy, các mô hình mạng tích chập đã có những chuyển biến không ngừng để cải thiện hiệu quả mô hình hóa ảnh kĩ thuật số cho các tác vụ Thị giác Máy tính.

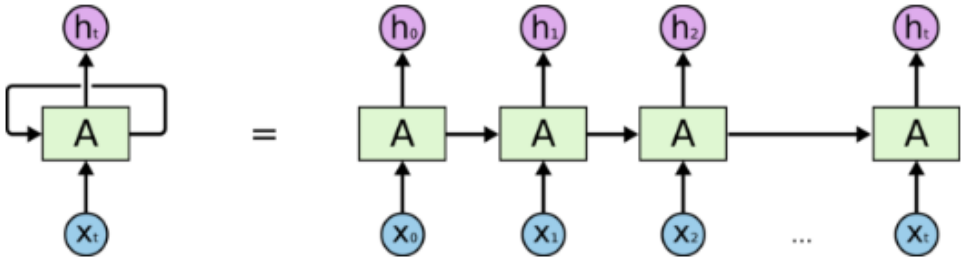
2.4. Mô hình mạng nơ-ron hồi quy cho biểu diễn văn bản

Trong mạng nơ-ron truyền thẳng cổ điển, ta giả định rằng các đầu vào là độc lập với nhau, tuy nhiên với nhiều tác vụ thì phép mô hình hóa này không phù hợp, giả sử khi mô hình hóa ngôn ngữ hay tín hiệu âm thanh tại mỗi thời điểm chẳng hạn. Mạng nơ-ron kết nối đầy đủ thông thường không phù hợp để mô hình hóa chuỗi do có số nút đầu vào cố định và độc lập với nhau, không mang tính tuần tự.

2.4.1. Mạng nơ-ron hồi quy

Mạng nơ-ron hồi quy (hay Recurrent Neural Network, viết tắt: RNN) là một mạng nơ-ron thường dùng trong các bài toán về mô hình hóa chuỗi, điển hình như:

Nhận diện giọng nói (*Speech Recognition*), **Dịch máy** (*Machine Translation*) hay **Sinh giọng nói từ văn bản** (*Text-to-speech*).



Để mô hình hóa một chuỗi x$x = x^{<1>}, x^{<2>}, \dots, x^{<N>}$, ta lần lượt truyền $x^{<t>}$ vào đơn vị **RNN** lần lượt tại mỗi thời điểm. Ở mỗi thời điểm, đơn vị **RNN** sẽ tồn tại một trạng thái kích hoạt ẩn $a^{<t>}$, trạng thái này sẽ được đưa vào làm thông tin cho trạng thái $(t+1)$ tiếp sau đó cùng với đầu vào $x^{<t+1>}$.

$$a^{<t>} = g(W_{aa}.a^{<t-1>} + W_{ax}.x^{<t>} + b_a) \quad (10)$$

$$g^{<t>} = g\left(W_{ya}.a^{<t>} + b_y\right) \quad (11)$$

Để thuận lợi cho việc tính toán, Ta có thể rút gọn theo cách sau:

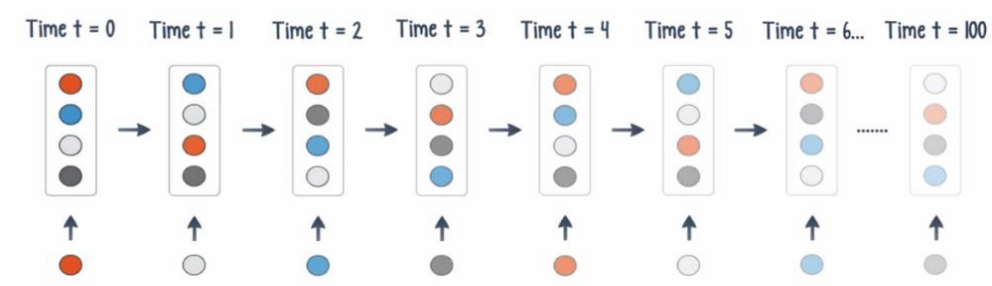
$$W_a = [W_{aa}; W_{ax}] = [W_{aa} \quad W_{ax}] \quad (12)$$

$$[a^{<t-1>}, x^{<t>}] = \begin{bmatrix} a^{<t-1>} \\ x^{<t>} \end{bmatrix} \quad (13)$$

Khi đó, ta có thể viết biểu thức $a^{<t>}$ lại thành:

$$a^{<t>} = g(W_a * [a^{<t-1>}, x^{<t>}] + b_a) \quad (14)$$

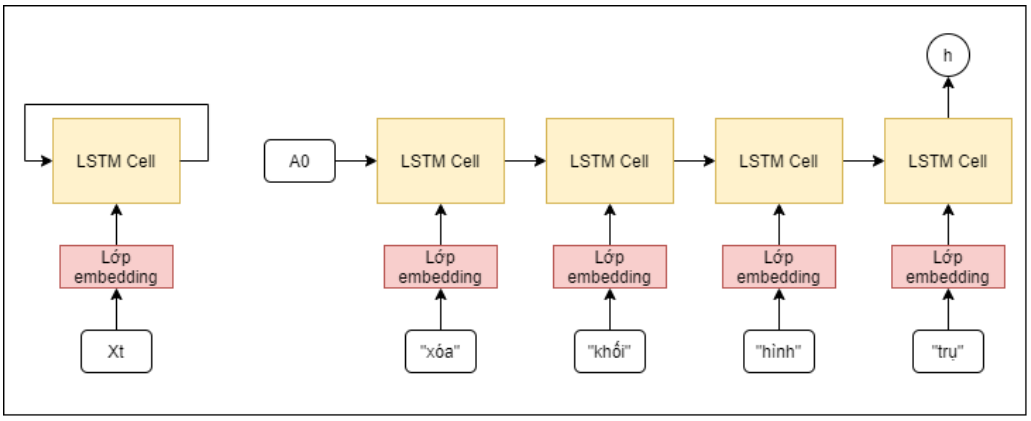
Mạng nơ-ron RNN là bước khởi đầu tốt cho một mô hình có thể biểu diễn chuỗi, tuy nhiên RNN lại xảy ra hiện tượng mất mát thông tin khi chúng ta mô hình hóa một chuỗi dài. Thông tin từ những thời điểm ban đầu sẽ bị bị ghi đè lên bởi thông tin tiếp theo, do đó chỉ có những thời điểm cuối cùng là chứa lượng thông tin dày đặc nhất.



Ảnh 2.20. Mất mát thông tin trong mạng nơ-ron hồi quy

2.4.2. Mạng bộ nhớ ngắn dài

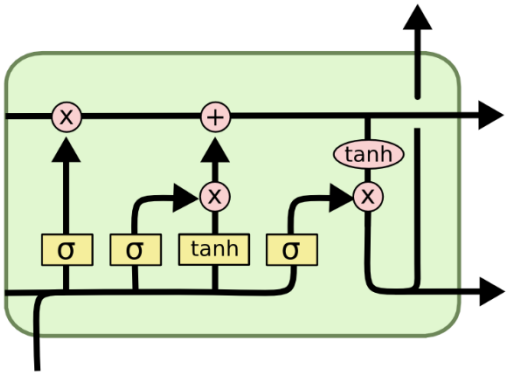
Mạng bộ nhớ ngắn dài (hay Long Short Term Memory, viết tắt: LSTM) là một biến thể của mạng hồi quy thông thường, với đặc thù kiến trúc giúp giảm thiểu sự *mất mát thông tin* nhờ cơ chế cho phép "nhớ" và "quên" thông tin.



Ảnh 2.21. Mạng LSTM sử dụng Embedding Layout cho biểu diễn từ trong mô hình của chúng tôi

Một đơn vị hồi quy trong LSTM được biểu diễn như sau, với đầu vào là trạng thái ẩn trước đó h_{t-1} và đầu vào tại thời điểm t là x_t.

$$
\begin{aligned}
i_t &= \sigma(x_t U^i + h_{t-1} W^i) \\
f_t &= \sigma(x_t U^f + h_{t-1} W^f) \\
o_t &= \sigma(x_t U^o + h_{t-1} W^o) \\
\widetilde{C_t} &= \tanh(x_t U^g + h_{t-1} W^g) \\
C_t &= \sigma\left(f_t * C_{t-1} + i_t * \widetilde{C_t}\right) \\
h_t &= \tanh(C_t) * o_t
\end{aligned}
\tag{15}
$$



Ảnh 2.22. Một đơn vị LSTM

Để chống lại hiện tượng *tiêu biến thông tin*, **LSTM** cho phép một *cơ chế nhớ* thông qua bộ nhớ $\boldsymbol{C_t}$ để thông tin có thể được *thêm vào* và *giữ lại* tại mỗi thời điểm. $\boldsymbol{i_t}$ (hay cổng thêm- insert gate) thiết lập một lớp mặt nạ (mask) biểu diễn cho tỉ lệ lượng thông tin mới được thêm vào. $\boldsymbol{f_t}$ (hay cổng quên – forget gate) thiết lập một lớp mặt nạ biểu diễn cho tỉ lượng thông tin được giữ lại so từ trạng thái $(\boldsymbol{t-1})$ trước đó. Sau đó **LSTM** thiết lập một ứng viên nhớ mới $\widetilde{\boldsymbol{C_t}}$, để được lấy trung bình với bộ nhớ tại thời điểm trước đó $\boldsymbol{C_t}$ với tham số tỉ lệ là $\boldsymbol{f_t}$ và $\boldsymbol{i_t}$.

2.5. Tiền xử lý dữ liệu

Ở phần này chúng tôi sẽ giới thiệu hai phương pháp tiền xử lí dữ liệu là **chuẩn hóa dữ liệu** và **tách từ**. Đây là một bước đệm quan trọng khi huấn luyện một mô hình *Học Máy.*

2.5.1. Chuẩn hóa dữ liệu

Chuẩn hóa dữ liệu là một phương thức tiền xử lí dữ liệu để phục vụ cho quá trình *Học Máy* được hiệu quả hơn.

Có khá nhiều phương pháp chuẩn hóa dữ liệu. Giả sử bạn có tập dữ liệu X, gồm N dòng và D cột (đặc trưng).

Theo lý thuyết, hồi quy tuyến tính sẽ không bị ảnh hưởng bởi chuẩn hóa dữ liệu bởi bất kì phép *biến đổi tuyến tính* của dữ liệu đầu vào đều có thể giải quyết bằng cách biến đổi bộ trọng số đầu vào một cách tuyến tính. Giả sử ta có một mô hình tuyến tính với bộ trọng số W và B.

$$Y = X * W + B \tag{16}$$

Trong đó $Y \in R^{N\times 1}, X \in R^{N\times D}$. Phép chuẩn hóa theo cột có thể biểu diễn bằng cách trừ với một ma trận M và nhân với ma trận đường chéo T

Khi đó:

$$\hat{X} = (X - M) * T = \begin{bmatrix} \frac{(x_{11} - m_1)}{t_1} & \cdots & \frac{(x_{1d} - m_d)}{t_d} \\ \frac{(x_{21} - m_1)}{t_1} & \cdots & \frac{(x_{2d} - m_d)}{t_1} \\ \vdots & \cdots & \vdots \\ \frac{x_{n1} - m_1}{t_1} & \cdots & \frac{(x_{nd} - m_1)}{t_d} \end{bmatrix} \tag{17}$$

Có thể thấy chúng ta có thể đối ứng với phép biến đổi này bằng cách thay đổi W và B.

$$Y = \hat{X} * \hat{W} + \hat{B} \tag{18}$$

Trong đó $\widehat{W} = T^{-1}W$ và $\hat{B} = B + M * W$. Vì lí do này, chuẩn hóa đầu vào sẽ không ảnh hưởng tới đầu ra hay độ chính xác của mô hình.

Tác dụng đầu tiên có thể kể đến của chuẩn hóa dữ liệu là tăng *tính ổn định tính toán* (numerical stability) khi huấn luyện mô hình. Nếu chúng ta có một tập dữ liệu X 1 chiều và chúng ta sử dụng **Mean Square Error** làm hàm độ lỗi, chúng ta cập nhật trọng số theo **Gradient Descent** như sau:

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial Y'} * \frac{\partial Y'}{\partial W} = \frac{2(Y - Y')^T}{N} * X \tag{19}$$

X giá trị càng lớn, khoảng cách từ trọng số khởi tạo W (mà được chọn ngẫu nhiên) và giá trị cực tiểu toàn cục sẽ rất nhỏ. Do đó với **learning rate** cố định, thuật toán **Gradient Descent** sẽ không thể hội tụ khi X lớn. Điều này sẽ làm cho hàm độ lỗi dao động và bùng nổ.

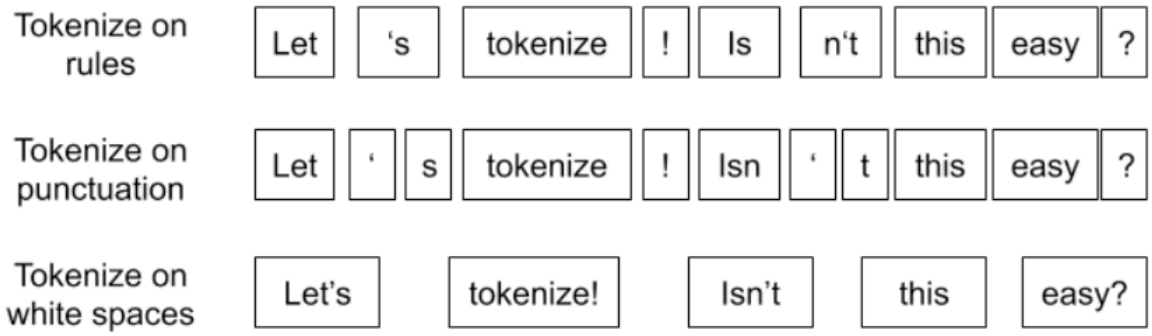
Chuẩn hóa dữ liệu có thể rất hữu ích trong nhiều trường hợp, có thể tăng tính ổn định tính toán và có thể tăng tốc độ huấn luyện. Tuy nhiên nó có thể không tốt đối với một số bài toán dựa trên khoảng cách, ví dụ: gom cụm dựa trên khoảng cách hay so khớp dữ liệu..

2.5.2. Tách từ

Xử lí ngôn ngữ tự nhiên là một lĩnh vực mà trong đó ngôn ngữ được xử lí bởi các chương trình phần mềm. XLNNTN có rất nhiều ứng dụng như ***phân tích cảm xúc*** (*sentiment analysis*), ***dịch máy*** (*machine translation*) hay ***nhận diện lỗi chính tả*** (*grammatical error detection*),…

Loại dữ liệu mà XLNNTN tiếp cận là dữ liệu văn bản, và dữ liệu này có thể đến bất kì đâu và khối lượng của chúng cực kì khổng lồ. Do vậy, những dữ liệu văn bản này cần được xử lí và làm sạch trước khi chúng có thể được dùng cho các tác vụ phân tích và thống kê. Mà trong đó, tách từ là một phép xử lí tiêu biểu cho văn bản.

Tách từ là chuyển một chuỗi thành một *tập hợp các token* có thứ tự, mà trong đó một token là một đơn vị ngữ nghĩa trong xử lý ngôn ngữ và mang một ý nghĩa cụ thể.



Ảnh 2.23. Ảnh minh họa về Tokenzation

Việc tách từ có thể được thực hiện theo nhiều cách khác nhau, tùy vào mục đích sử dụng của người thiết kế thuật toán tách từ đó. Một số cách tách từ phổ biến như là tách từ dựa trên dấu chấm câu hay dựa trên khoảng trắng trong câu.

Chương 3. XÂY DỰNG TẬP DỮ LIỆU TIẾNG VIỆT

3.1. Xây dựng công cụ dịch sử dụng cây cú pháp

Trong những năm gần đây, đã xuất hiện rất nhiều tập dữ liệu tiếng Việt được xây dựng dựa trên các tập dữ liệu tiếng Anh. Tuy nhiên, phương pháp dịch của các bạn còn gặp nhiều hạn chế, thủ công và chưa thật sự hiệu quả. Do đó chúng tôi quyết định xây dựng một quy trình dịch tự động, được đóng gói thành một bộ công cụ dịch để có thể tái sử dụng.

3.1.1. Một số hướng tiếp cận dịch dữ liệu hiện tại

Khi chọn dịch một tập dữ liệu, thường các nghiên cứu viên sẽ chọn một trong hai hướng tiếp cận sau.

Hướng tiếp cận đầu tiên là xây dựng một đội ngũ dịch gồm một lượng nhân lực nhất định. Hướng tiếp cận này rất hiệu quả đối với các dự án cần tính tỉ mỉ cao và độ chính xác của con người. Các nhóm dịch có chuyên môn cao thường sẽ cung cấp những hướng dẫn dịch cụ thể và có những chuyên gia đối soát lại kết quả dịch. Tuy nhiên, khuyết điểm lớn nhất của phương pháp này là rất tốn kém, đòi hỏi nguồn nhân lực nhiều và rất khó rà soát lỗi dịch nếu có lỗi xảy ra. Ngoài ra, việc dịch còn bị ảnh hưởng bởi tính chủ quan của người tham gia dịch.

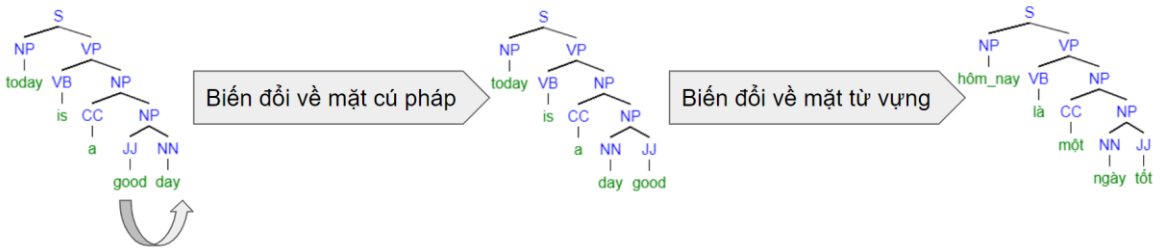
Hướng tiếp cận thứ hai là sử dụng một phần mềm dịch máy (Như Google Translate) hoặc mô hình dịch máy do bên thứ ba phát triển. Hướng tiếp cận này rất tiết kiệm chi phí vì việc dịch được diễn ra tự động hoàn toàn. Tuy nhiên, kết quả dịch thường sẽ không thể kiểm soát được và tồn tại nhiều sai sót dù sao đây cũng là một mô hình dịch máy.

Hướng tiếp cận thứ ba là kết hợp sử dụng một mô hình học máy để dịch và sau đó kiểm tra lại bằng sức người. Phương pháp này cải thiện nhược điểm của cả

hai phương pháp trên như chi phí nhân công cao, đồng thời, kết quả được trả về hoàn toàn tự động.

3.1.2. Ý tưởng của công cụ dịch dựa trên cây cú pháp

Từ những hạn chế kể trên, chúng tôi đi đến việc xây dựng bộ công cụ dịch tự động dựa trên tập luật **URBANS** (**U**niversal ***R**ule-**ba**sed Tra**ns**lation toolkit*), với mong muốn giải quyết các vấn đề được nêu ở trên.



Ảnh 3.1. Minh họa quy trình dịch của công cụ dịch dựa trên tập luật URBANS

Ý tưởng chính của công cụ dịch dựa trên cây cú pháp là dịch bằng cách di chuyển các nút trên cây trên cây cú pháp và ánh xạ từ từ ngôn ngữ gốc tới ngôn ngữ đích.

Đầu tiên chúng sẽ thực hiện biến đổi về mặt cú pháp bằng cách di chuyển các nút trên cây cú pháp của ngôn ngữ gốc. Ví dụ: Ở tiếng Anh tính từ nằm trước danh từ còn ở tiếng Việt thì ngược lại. Do đó khi thực hiện dịch từ tiếng Anh sang tiếng Việt, ta biến đổi về mặt cú pháp bằng cách thay đổi vị trí của tính từ và động từ.

Sau đó chúng ta thực hiện biến đổi về mặt từ vựng bằng cách sử dụng ánh xạ một-một từ ngôn ngữ gốc tới ngôn ngữ đích. Ví dụ: “dog” ở tiếng Anh ánh xạ sang “chó” ở tiếng Việt, “a” của tiếng Anh ánh xạ sang “một” ở tiếng Việt.

3.1.3. Những ưu và nhược điểm

3.1.3.1. Những ưu điểm

Công cụ dịch dựa trên cây cú pháp **URBANS** [5] có một số ưu điểm nổi bật. Đầu tiên là cho phép chúng ta *điều khiển kết quả đầu ra*, là ưu điểm không thể có được nếu ta sử dụng một mô hình học máy hay các ứng dụng Google Translate để dịch tập dữ liệu. Công cụ dịch này còn cho phép chúng ta *quản lí quy trình dịch* một cách khoa học và hiệu quả, mọi thay đổi mong muốn sẽ được thực hiện trên tập luật thay vì trực tiếp trên văn bản cần dịch. Khi có bất kì lỗi gì được phát sinh từ quá trình dịch, chúng ta có thể chỉnh sửa nhanh chóng trên tập luật mà không cần phải truy xét lại toàn bộ các văn bản cần dịch.

3.1.3.2. Những mặt hạn chế

Ngoài những ưu điểm kể trên, công cụ dịch không thể tránh khỏi một số thiếu sót nhất định. Một trong những hạn chế có thể kể đến là công cụ dịch chưa hỗ trợ chỉnh sửa trên cây cú pháp có ràng buộc ngữ nghĩa . Ngoài ra, việc sử dụng công cụ dịch này cũng đòi hỏi người dùng phải có hiểu biết nhất định về việc phân tích cú pháp trong Xử lý ngôn ngữ tự nhiên, đồng thời phải biết về cấu trúc ngữ pháp mà về tập dữ liệu mà mình muốn dịch.

3.1.4. Cách cài đặt và minh họa

Việc cài đặt công cụ này trên môi trường lập trình Python rất đơn giản thông qua *bash terminal*:

```
pip install URBANS
```

Ở ví dụ demo này, chúng tôi sẽ dịch 2 câu tiếng Anh sau thành tiếng Việt sử dụng công cụ **URBANS** [5].

```
from URBANS import Translator

# Một số câu cần dịch ở ngôn ngữ gốc
src_sentences = ["I love good dogs", "I hate bad dogs"]

# Cấu trúc ngữ pháp của ngôn ngữ gốc theo định dạng của thư viện nltk
src_grammar = """
                S -> NP VP
                NP -> PRP
                VP -> VB NP
                NP -> JJ NN
                PRP -> 'I'
                VB -> 'love' | 'hate'
                JJ -> 'good' | 'bad'
                NN -> 'dogs'
                """

# Một số thay đổi từ ngôn ngữ tiếng Anh tới ngôn ngữ tiếng Việt
src_to_target_grammar =  {
    "NP -> JJ NN": "NP -> NN JJ" # Ở tiếng Việt danh từ đi trước tính từ
}

# Từ điển ánh xạ một-một giữa tiếng Anh và tiếng Việt
en_to_vi_dict = {
    "I":"tôi",
```

```
    "love":"yêu",
    "hate":"ghét",
    "dogs":"những chú_chó",
    "good":"ngoan",
    "bad":"hư"
    }

translator = Translator(
            src_grammar = src_grammar,
            src_to_tgt_grammar = src_to_target_grammar,
            src_to_tgt_dictionary = en_to_vi_dict
          )

# Thực hiện dịch
trans_sentences = translator.translate(src_sentences)

print(trans_sentences)
```

Output:

```
['tôi yêu những chú_chó ngoan', 'tôi ghét những chú_chó hư']
```

3.2. Xây dựng tập dữ liệu CSS-VN

Tập dữ liệu **CSS-VN** được xây dựng bằng việc thay thế *câu mô tả tăng cường* trong tập dữ liệu **CSS** tiếng Anh bằng bản dịch của nó trong tiếng Việt. Việc dịch **CSS** tập dữ liệu **CSS** được thực hiện bằng cách sử dụng công cụ dịch tự động dựa trên tập luật **URBANS** [5] kể trên.

3.2.1. Phân tích sơ bộ tập dữ liệu CSS

Chúng tôi tiến hành phân tích sơ bộ tập dữ liệu CSS thì thấy rằng tập dữ liệu CSS có **15** cấu trúc ngữ pháp cơ sở như sau:

Loại	Cấu trúc ngữ pháp cơ sở	Ví dụ
1	VB JJ_pos JJ NN JJ	make bottom-left gray circle blue
2	VB JJ_pos NN JJ	make top-left rectangle blue
3	VB JJ NN JJ	make brown circle red
4	VB JJ NN	remove red circle
5	VB JJ NN PP JJ_pos	add red triangle to bottom-left
6	VB NN JJ	make triangle yellow
7	VB JJ_pos NN	remove middle-right circle
8	VB JJ_pos JJ NN	remove middle-center gray triangle
9	VB NN PP JJ_pos	add triangle to top-center
10	VB NN	remove circle
11	VB JJ JJ NN	remove large blue triangle
12	VB JJ JJ NN JJ	make small red circle yellow
13	VB JJ_pos JJ JJ NN	remove middle-left small cyan rectangle
14	VB JJ_pos JJ JJ NN JJ	make top-center large gray object green
15	VB JJ JJ NN PP JJ_pos	add large blue circle to middle-right

Bảng 3.1. Phân tích sơ bộ cấu trúc ngữ pháp của tập dữ liệu CSS

Có thể thấy, tập dữ liệu có cấu trúc ngữ pháp khá đơn điệu, khoảng 16 cấu trúc ngữ pháp cơ sở. Trong đó, ta có một số từ loại như sau.

Từ khóa	Từ loại	Danh sách từ	Số lượng
VB	Động từ	make	3
		add	
		remove	
NN	Danh từ	object	4
		sphere	
		cyclinder	
		cube	
JJ	Tính từ	brown	19
		green	
		blue	
		gray	
		purple	
		cyan	
		red	
		yellow	
		middle-left	
		middle-right	
		middle-center	
		bottom-center	
		bottom-left	
		bottom-right	
		top-right	
		top-left	
		top-center	
		small	
		large	

PP	Giới từ	to	1

3.2.2. Tiến hành tập dữ liệu CSS dựa theo tập luật

Chúng tôi tiến hành dịch tập dữ liệu CSS dựa trên tập luật, dựa trên công cụ **URBANS** như sau.

Đầu tiên, chúng tôi thực hiện **biến đổi về mặt ngữ pháp** dựa trên cây cú pháp. Việc chuyển đổi này từ tiếng Anh sang tiếng Việt rất đơn giản, do tính đơn điệu của tập dữ liệu CSS. Sau đó, chúng tôi biến đổi về mặt từ vựng bằng một phép **ánh xạ một-một** (*one-to-one mapping*) từ ngôn ngữ tiếng Anh sang ngôn ngữ tiếng Việt.

3.2.2.1. Biến đổi về mặt ngữ pháp

Đầu tiên, chúng ta sẽ thực hiện *biến đổi về mặt ngữ pháp*.

Ngữ pháp gốc (tiếng Anh)	Ngữ pháp đích (tiếng Việt)
`NP -> JJ NP`	`NP -> NP JJ`
`VP -> VP JJ`	`VP -> VP (PP thành) JJ`

Bảng 3.2. Biến đổi về mặt cú pháp

Ở tiếng Anh, tính từ `JJ` đứng trước danh ngữ `NP` còn ở tiếng Việt thì ngược lại. Khi thực hiện phép biến đổi chúng ta sẽ thay đổi vị trí của hai thành phần này.

3.2.2.2. Biến đổi về mặt từ vựng

Tiếp theo, chúng ta thực hiện biến đổi câu về mặt từ vựng theo phép **ánh xạ một-một** (*one-to-one mapping*).

Từ khóa	Từ loại	Ngôn ngữ gốc	Ngôn ngữ đích
VB	Động từ	make	Biến
		add	Thêm
		remove	Xóa

NN	Danh từ	object	Khối
		sphere	Khối cầu
		cylinder	Khối trụ
		cube	Khối lập phương
JJ	Tính từ	brown	màu nâu
		green	màu xanh lá
		blue	màu xanh dương
		gray	màu xám
		purple	màu tím
		cyan	màu lam
		red	màu đỏ
		yellow	màu vàng
		middle-left	bên trái
		middle-right	bên phải
		middle-center	trung tâm
		bottom-center	chính giữa phía dưới
		bottom-left	góc trái phía dưới
		bottom-right	góc phải phía trên
		top-right	góc phải phía trên
		top-left	góc trái phía trên
		top-center	chính giữa phía trên
		small	nhỏ
		large	lớn
PP	Giới từ	to	vào

Bảng 3.3. Ánh xạ từ vựng một-một khi dịch văn bản

3.2.3. Một số kết quả dịch mẫu

Bên dưới là một số kết quả dịch mẫu ứng với 15 loại câu

Loại câu	Tiếng Anh	Tiếng Việt
1	make bottom-left gray sphere blue	biến khối cầu màu xám góc trái phía dưới thành màu xanh dương
2	make top-left cube blue	biến khối hộp chữ nhật góc trái phía trên thành màu xanh dương
3	make brown sphere red	biến khối cầu màu nâu thành màu đỏ
4	remove red sphere	xóa khối cầu màu đỏ
5	add red cylinder to bottom-left	thêm khối trụ màu đỏ vào góc trái phía dưới
6	make cylinder yellow	biến khối trụ thành màu vàn
7	remove middle-right sphere	xóa khối cầu bên phải
8	remove middle-center gray cylinder	xóa khối trụ màu xám trung tâm
9	add cylinder to top-center	thêm khối trụ vào chính giữa phía trên
10	remove sphere	xóa khối cầu
11	remove large blue cylinder	xóa khối trụ màu xanh dương lớn
12	make small red sphere yellow	biến khối cầu màu đỏ nhỏ thành màu vàng

13	remove middle-left small cyan cube	xóa khối hộp chữ nhật màu lam nhỏ bên trái
14	make top-center large gray object green	biến khối màu xám lớn chính giữa phía trên thành màu xanh lá
15	add large blue sphere to middle-right	Thêm khối cầu màu xanh dương vào bên phải

Bảng 3.4. Một số kết quả dịch mẫu dựa trên cây cú pháp

Chương 4. TEXT-IMAGE RESIDUAL GATING CHO KẾT HỢP ẢNH VÀ CÂU MÔ TẢ TĂNG CƯỜNG TIẾNG VIỆT ĐỂ TRUY VẤN ẢNH

4.1. Giới thiệu

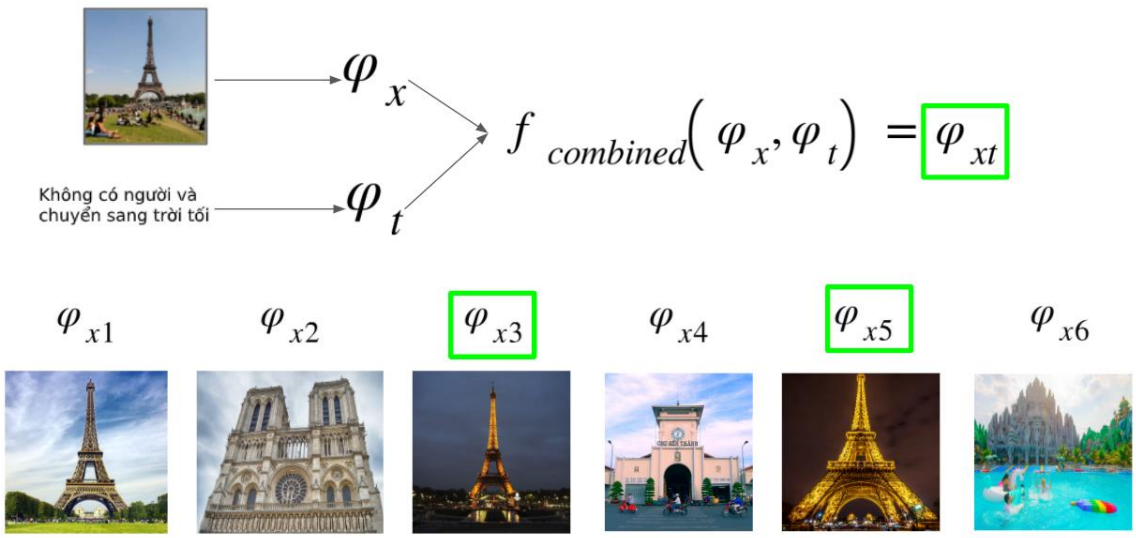
Bài báo *"Composing text and image for image retrieval"* [1] nghiên cứu về bài *Truy vấn Thông tin*, mà trong đó truy vấn đầu vào gồm **hai thành phần** là *ảnh tham khảo* và *văn bản tăng cường*, với văn bản tăng cường mô tả sự thay đổi mong muốn đối với tấm ảnh tham khảo. Qua đó tác giả đề xuất là một mô hình học sâu biểu diễn kết hợp cặp truy vấn đầu vào là ảnh và văn bản, gọi là **Text-Image Residual Gating** (*TIRG*). Mục tiêu của **TIRG** là tìm ra một không gian biểu diễn cho câu truy vấn kết hợp ảnh và câu mô tả tăng cường, để sử dụng biểu diễn này để truy vấn trong cơ sở dữ liệu ảnh sao cho biểu diễn câu truy vấn (ảnh + văn bản) và ảnh mục tiêu gần nhau trên không gian biểu diễn. Phương pháp này vượt trội hơn so với các phương pháp hiện tại trên 3 tập dữ liệu khác nhau, bao gồm: **Fashion-200k**, ***MIT-States*** và một bộ dữ liệu được tổng hợp dựa trên CLEVR là **CSS**. Bài báo cũng chứng minh rằng phương pháp **TIRG** cũng có thể được sử dụng để phân loại dựa với loại đặc trưng kết hợp được đề xuất trong bài báo, và vượt trội hơn tập dữ liệu *MIT-States* trên tác vụ này.

Để giải quyết bài truy vấn kết hợp ảnh và câu mô tả tăng cường cho truy vấn ảnh, chúng tôi chuyển bài toán trên thành một bài toán chuyển đổi như sau.

Trong không gian vector, cho trước:

- Một tấm ảnh tham khảo x và một câu truy vấn tăng cường t, với biểu diễn ϕ_x và ϕ_t tương ứng.
- Một cơ sở dữ liệu ảnh gồm n ảnh, với các biểu diễn lần lượt là $\phi_{x_1}, \phi_{x_2}, \dots, \phi_{x_n}$

Nhiệm vụ của chúng ta là xây dựng một hàm $f_{kết\ hợp}: \phi_x, \phi_t \rightarrow \phi_{xt}$ ánh xạ từ biểu diễn của ảnh ϕ_x và và câu mô tả tăng cường ϕ_t sang một biểu diễn chung cho cặp ảnh – câu mô tả là ϕ_{xt}, từ đó ta có thể sử dụng biểu diễn kết hợp ϕ_{xt} để truy vấn trong cơ sở dữ liệu ảnh.

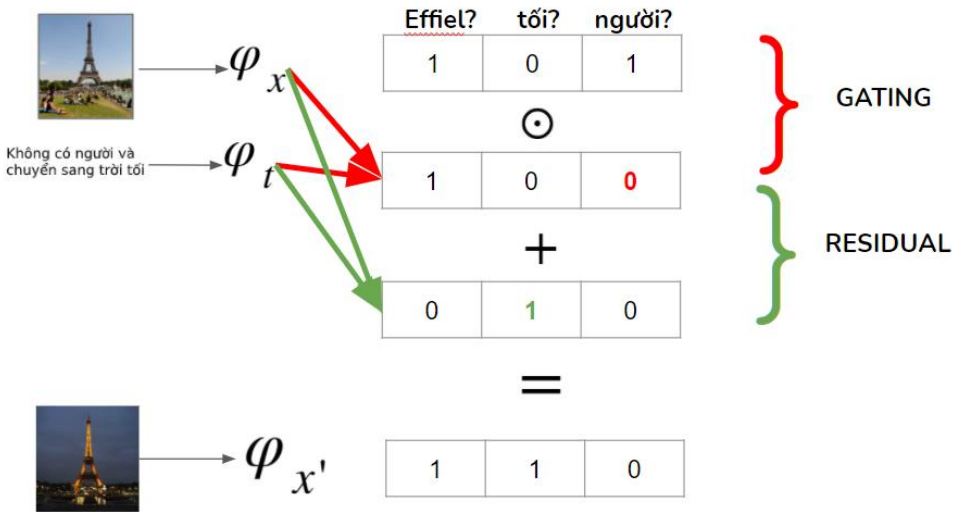


Ảnh 4.1. Ảnh minh họa về hướng tiếp cận cho bài toán truy vấn ảnh sử dụng ảnh và câu mô tả tăng cường

4.2. Phương pháp

4.2.1. Ý tưởng chính

Ý tưởng chính của bài toán là chúng ta có thể biến đổi ϕ_x bất kì thành vector khác thông qua 2 phép toán là nhân Hadamard $\odot$ và cộng ma trận.



Ảnh 4.2. Biến đổi vectơ sử dụng phép nhân Hadamard và cộng ma trận

Cho vector $\phi_x = [x_1, x_2, \dots, x_d] \in R^d$. Ta có thể hiểu mỗi thành phần x_i ($i \in \{x \in N \mid x \leq d\}$) là một "khái niệm". Ta có thể điều chỉnh vector ϕ_x thành vector ϕ'_x thông qua một hàm biến đổi $f_{\text{biến đổi}}$ gồm 2 bước sau:

- **Gating**: Ta áp một lớp mặt nạ (mask) $m = [m_1, m_2, \dots, m_d] \in [0,1]^d$ để có thể hãm hoặc giữ lại một số thành phần trong vector.
- **Residual**: Ta cộng với một vector dự (residual vector) $r = [r_1, r_2, \dots, r_d] \in R^+$ để có tăng giá trị của một số thành phần trong vector.

$$\phi'_x = f_{\text{biến đổi}}(\phi_x, m, r) = \phi_x \odot m + r \tag{20}$$

Để phép biểu diễn ϕ'_x gần với phép biểu diễn của ảnh mục tiêu ϕ_{x_target}, ta thiết lập một bài toán tối ưu với m, r được tham số hóa, như sau:

$$L = \underset{m,r}{\operatorname{argmin}}\, d(\phi'_x, \phi_{x_{target}}) - d(\phi'_x\, \phi_{x_{nontarget}}) \tag{21}$$

Trong đó $d(\phi_1, \phi_2)$ là một phép đo khoảng cách bất kì nhận vào ϕ_1, ϕ_2.

4.2.2. Text-Image Residual Gating

Đầu tiên, cho trước một tấm ảnh x, chúng tôi sử dụng mạng tích chập **ResNet-17** để rút trích các đặc trưng về không gian 2D của ảnh, $f_{ảnh}(x) = \phi_x \in R^{WxHxC}$ trong đó W, H và $C = 512$ lần lượt chiều dài, chiều rộng và chiều cao của bản đồ đặc trưng của tấm ảnh . Sau đó, chúng tôi biểu diễn câu *mô tả tăng cường t* sử dụng mạng **LSTM**. Khi đó biểu diễn của t sẽ là $f_{text}(t) = \phi_t \in R^d$ là đầu ra của LSTM tại thời điểm cuối cùng. Sau đó chúng ta sẽ kết hợp ảnh và câu mô tả tăng cường thành một biểu diễn đồng nhất gọi là $\phi_{xt} = f_{kết\ hợp}(\phi_x, \phi_t)$.

Bài báo [1] đề xuất một phép một cách thiết lập phép kết hợp $f_{kết\ hợp}$, gọi là **Text-Image Residual Gating** (TIRG), với:

$$f_{kết\ hợp}: (\phi_x, \phi_t) \rightarrow \phi_{xt} \tag{22}$$

Với hàm $f_{kết\ hợp}$ được định nghĩa như sau:

$$\left(f_{kết\ hợp}\right):\ \phi_{xt} = \ w_g f_{gate}(\phi_x, \phi_t) + w_r f_{res}(\phi_x, \phi_t) \tag{23}$$

Trong đó f_{gate} và $f_{res} \in R^{W \times H \times C}$ là đặc trưng gating và residual được giới thiệu ở Ảnh 8. Các tham số học được là w_g và w_r để cân bằng giá trị đầu ra của f_{gate} và f_{res}.

Hàm f_{gate} được tính toán như sau:

$$f_{gate} = \sigma(W_{g2} *\ Relu\left(W_{g2} * [\phi_x, \phi_t]\right)) \odot \phi_x \tag{24}$$

Trong đó, σ là hàm sigmoid, $\odot$ là phép nhân **element-wise** (hoặc phép nhân Hadamard), $*$ là phép tích chập với **Batch Normalization** với W_{g1} và W_{g2} là bộ lọc 3x3. Khi thực hiện phép **concatenate** $[\phi_x, \phi_t]$, vectơ ϕ_t được broadcast để đảm bảo rằng nó cùng kích thước với ϕ_x ở chiều thứ 3. Việc sử dụng hàm sigmoid là để xây dựng một lớp mặt nạ (mask) với các giá trị được rang buộc trong

khoảng (0,1), cho phép hãm hoặc giữ lại giá trị mong muốn của ϕ_x thông qua phép nhân Hadamard $\odot$.

Hàm f_{res} được tính toán như sau:

$$f_{res} = W_{r2} * Relu(W_{r1} * [\phi_x, \phi_t]) \quad (25)$$

Mục tiêu của hàm f_{res} là học ra một lớp *residual feature* (đặc trưng thặng dư) để biểu diễn một số đặc trưng muốn thêm vào ϕ_x sau khi hãm hay giữ giá trị của ϕ_x từ hàm f_{gate}. (Xem công thức 22)

4.2.3. Deep Metric Learning

Mục tiêu của chúng ta là kéo những biểu diễn của ϕ_{xt} gần với biểu diễn của ảnh mục tiêu ϕ_{x_target} lại gần với nhau, và kéo đặc trưng của ảnh không phải mục tiêu $\phi_{x\_nontarget}$ ra xa. Để huấn luyện mô hình trên, chúng tôi sử dụng **hàm lỗi phân lớp** (*classification loss*). Cụ thể, chúng tôi huấn luyện một minibatch gồm Q câu truy vấn, trong đó $\psi_i = f_{\text{kết hợp}}(x_i^{target}, t_i)$, là phép biểu diễn kết hợp ở lớp cưới cùng của câu truy vấn (ảnh, văn bản), và $\phi_i^+ = f_{img}(x_i^{target})$ là là biểu diễn của ảnh mục tiêu của câu truy vấn thứ i đó.

Với mỗi batch b, chúng tôi tạo một tập $\mathcal{N}_i$ bao gồm K mẩu: một mẩu ϕ_i^+ và $K-1$ mẩu negative $\phi_1^-, \dots, \phi_{K-1}^-$ (bằng cách lấy mẩu ϕ_j từ minibatch với $j \neq i$.)

$$L = \frac{1}{MB}\sum_{b}^{B}\sum_{m}^{M}\log\{\frac{\exp\{\mathcal{K}(\psi_i, \phi_i^+)\}}{\sum_{\phi_j \in \mathcal{N}_b^m}\exp\{\mathcal{K}(\psi_i, \phi_j)\}}\} \quad (26)$$

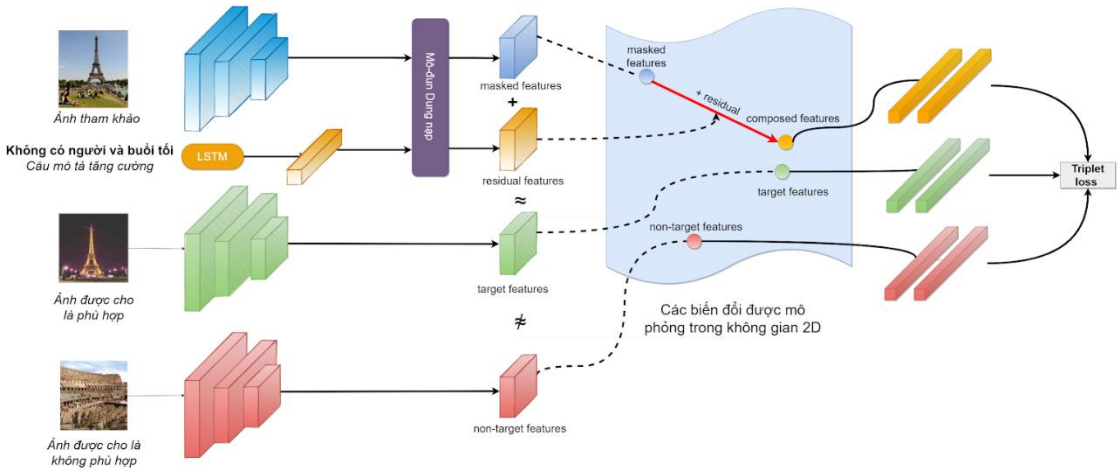
Trong đó $\mathcal{K}$ là **hàm tương đồng** (*similarity kernel/function*) lấy vào 2 vectơ. Trong phương pháp chúng tôi sử dụng thì $\mathcal{K}$ sử dụng **tích vô hướng** và **hàm đối *l*2** (*negative l*2).

Khi $K = 2$ chúng ta có thể viết lại hàm L trên thành:

$$L = \frac{1}{MB}\sum_{i}^{B}\sum_{m}^{M}\log\{\exp\{\mathcal{K}(\psi_i, \phi_i^+)\} - \exp\{\mathcal{K}\left(\psi_i, \phi_j^-\right)\}} \quad (27)$$

Có thể thấy hàm L (8) trên được viết lại đơn giản hơn rất nhiều bởi vì chỉ tồn tại một điểm dữ liệu negative. Hàm này tương đương với hàm **Soft Triplet Based Loss** được sử dụng trong [12, 25]. Khi sử dụng $K = 2$, chúng tôi sử dụng $M = B - 1$, để có thể ghép cặp mẩu i với tất cả các mẩu negative khác có thể.

Quá trình huấn luyện mạng **TIRG** được chúng tôi mô tả lại bằng hình bên dưới như sau:



Ảnh 4.3. Kiến trúc và quy trình huấn luyện của mạng **TIRG**

Chúng tôi huấn luyện mạng **TIRG** dựa trên **Mini-batch Gradient Descent** với các thành phần như sau, với mỗi mini-batch:

- Ảnh tham khảo được đi qua mô-đun kết hợp TIRG nhận phép biểu diễn $\psi_i = f_{\text{kết hợp}}\left(x_i^{target}, t_i\right)$
- Ảnh được cho là phù hợp và ảnh không phù hợp được cho qua mạng CNN với phép biểu diễn tương ứng là ϕ_i^+ và ϕ_i^-

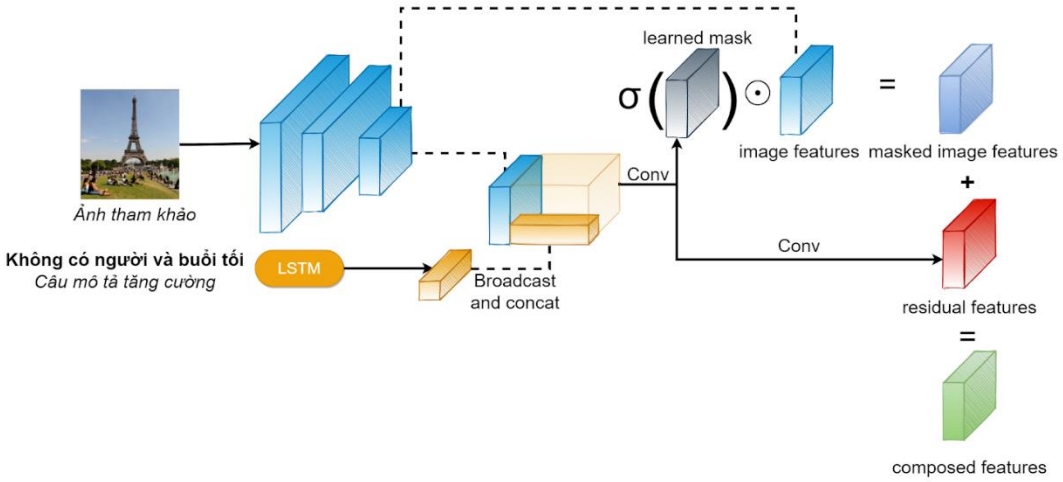
Chúng tôi tối ưu bằng cách tìm kiếm bộ trọng số sao cho cực tiểu hóa hàm L (25) với đầu vào là ψ_i, ϕ_i^+ và ϕ_i^- theo **Gradient Descent**:

$$\mathrm{w} = \underset{w}{\operatorname{argmin}} L \tag{28}$$

4.3. Hai cấu hình của mô hình Text-Image Residual Gating

Ngoài kiến trúc mô hình **TIRG** gốc, tác giả còn giới thiệu thêm một biến thể của **TIRG**. Điểm khác nhau của biến thể này so với phiên bản **TIRG** là mô-đun kết hợp được áp dụng ở đặc trưng đặc trưng lớp Fully Connected thay vì bản đồ đặc trưng (feature map) của lớp convolution. Tác giả cho rằng việc kết hợp ở các vị trí khác nhau sẽ phù hợp với các loại dữ liệu khác nhau.

4.3.1. Mô hình Text-Image Residual Gating với mô-đun kết hợp ở lớp Convolution



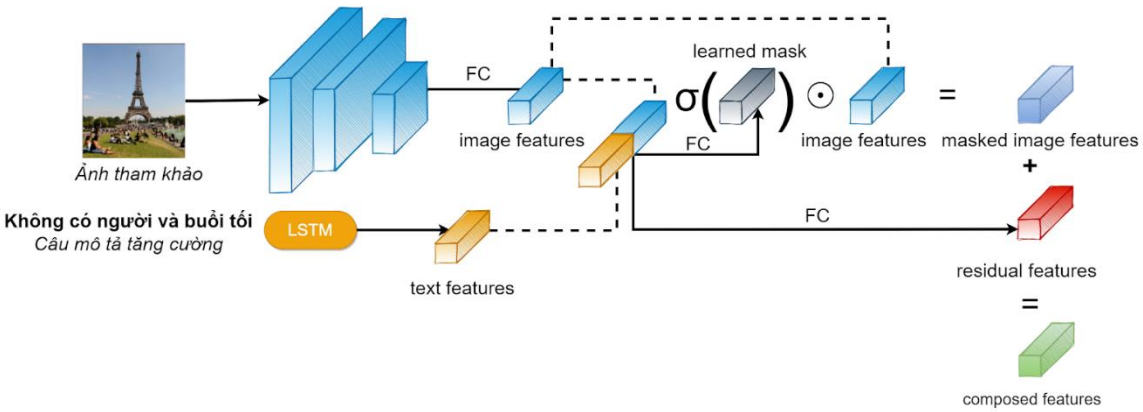
Ảnh 4.4. Mô hình TIRG với mô-đun kết hợp ở lớp Convolution

Đây là cấu hình mặc định của **TIRG** được giới thiệu trong bài báo. Ở trong mô hình đầu tiên này, việc thực hiện kết hợp sẽ xảy ra ở bản đồ đặc trưng của lớp convolution. Ở đây, ảnh tham khảo sẽ được biểu diễn bởi mạng **CNN**, câu mô tả tăng cường được biểu diễn bằng **LSTM.** Sau đó biểu diễn của câu mô tả được broadcast và chồng với biểu diễn của ảnh ở chiều thứ ba, để đảm bảo mỗi điểm

trên không gian bản đồ đặc trưng đều có thông tin của của câu văn bản khi thực hiện phép tích chập sau đó. Khối Tensor này sau đó sẽ được tích chập để sinh ra một vectơ mặt nạ (learned mask) và vectơ dư (residual) cho phép hãm hoặc thêm thông tin vào bản đồ đặc trưng ảnh trước đó.

Mô hình này sẽ phù hợp đối với những tập dữ liệu gồm những sự *biến đổi cục bộ*. Ví dụ như là tập dataset CSS. Đối với tập dữ liệu CSS, các biến đổi được thực hiện ở những vùng nhỏ trên bức ảnh. Ví dụ như "xóa khối cầu màu đỏ" hay "thêm khối trụ màu vàng". Do những biến đổi này là các *biến đổi cục bộ*, việc thực hiện biến đổi trên bản đồ đặc trưng của lớp Convolution sẽ phù hợp hơn do bản đồ đặc trưng này giữ được các thông tin về vị trí của bức ảnh.

4.3.2. Mô hình Text-Image Residual Gating với mô-đun kết hợp ở lớp Fully Connected



Ảnh 4.5. Mô hình TIRG với mô-đun kết hợp ở lớp Fully Connected

Ở trong phép kết hợp thứ hai, chúng ta thực hiện việc kết hợp ở đặc trưng của lớp *Fully Connected* thay vì lớp *Convolution* như trên. Để làm được việc này, ảnh sau khi được rút trích được bản đồ đặc trưng sau mạng CNN sẽ được truyền thẳng qua một lớp kết nối đầy đủ để biến thành một vectơ để chồng nối tiếp với

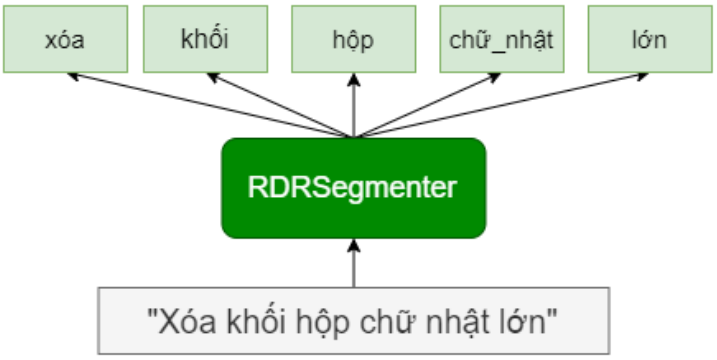
đặc trưng câu mô tả. Vectơ chồng này sau đó cũng được đem đi để học ra một vectơ mặt nạ và vectơ dự để dùng biến đổi đặc trưng ảnh.

Do đặc trưng của lớp *Fully Connected* là những đặc trưng mang tính *toàn cục*, bao hàm ngữ nghĩa và khái niệm của toàn bộ bức ảnh đầu vào, việc kết hợp ở đặc trưng *Fully Connected* sẽ phù hợp hơn với tập dữ liệu với các *biến đổi toàn cục*, điển hình như tập dữ liệu **MIT-States** hay **Fashion200k**. Trong 2 tập dữ liệu này, các biến đổi (được biểu diễn dưới câu mô tả tăng cường) là những biến đổi toàn cục, có khả năng thay đổi những đặc điểm về thị giác của toàn bức ảnh.

4.4. Giải quyết sự nhập nhằng khoảng trống sử dụng RDRSegmenter

Như ta đã biết, tiếng Việt dù có lợi thế khi mô hình hóa bằng mô hình Học máy do tính không có biến tố của nó. Tuy nhiên vẫn tồn đọng sự *nhập nhằng khoảng trống,* làm cho việc huấn luyện mô hình Học Máy không được hiệu quả

Để giải quyết vấn đề trên, chúng tôi sử dụng bộ tách từ **RDRSegmenter** để token-hóa các từ trong câu mô tả đầu vào thành một tập hợp các token, mà trong đó mỗi token là đơn vị nhỏ nhất mang ngữ nghĩa**.**



Ảnh 4.6. Sử dụng RDRSegmenter để tách từ

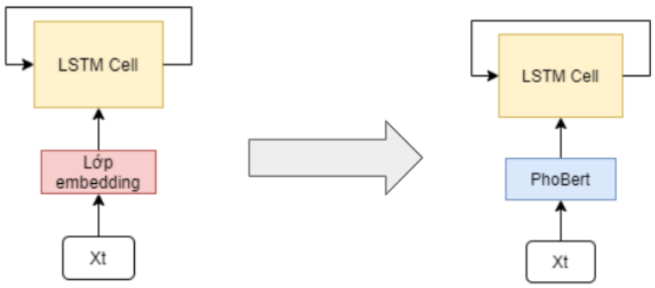
RDRSegmenter là một công cụ tách từ state-of-the-art ở thời điểm hiệu tại, vượt mặt **UETSegmenter, DongDu** hay cả **JVnSegmenter-CRFs.** Việc sử dụng

một công cụ tách từ sẽ giúp giảm thiểu nhập nhằng khoảng trắng ở tiếng Việt, giúp mô hình Học máy hoạt động hiệu quả hơn.

4.5. Thích ứng với dữ liệu mới bằng cách sử dụng PhoBERT làm bộ biểu diễn từ

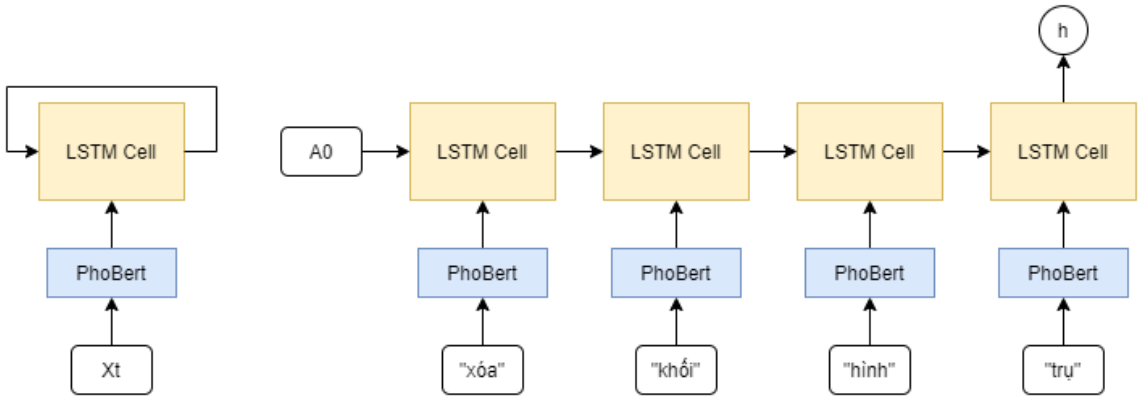
Tập dữ liệu **CSS-VN** (cũng như tập dữ liệu gốc ở tiếng Anh) có các câu mô tả với số từ vựng hạn chế, chỉ có khoảng 27 từ. Một số từ như động từ "xóa" có thể thay thế bằng "loại" hoặc "bỏ", tuy nhiên lại không có trong từ điển của lớp embedding của mạng **LSTM**. Nếu sử dụng một lớp embedding đối những từ *nằm ngoài từ điển* thì sẽ không tìm ra được phép biểu diễn phù hợp. Do đó chúng tôi quyết định sử dụng một bộ biểu diễn từ (và câu) tiếng Việt được tiền huấn luyện là **PhoBERT**, vốn được huấn luyện trên một tập dữ liệu khổng lồ gồm 20GB văn bản.

Bằng việc sử dụng **PhoBERT**, chúng ta sẽ *giảm chi phí tính toán* gradient ở bước lan truyền ngược cho *mô-đun biểu diễn từ* (ở đây là PhoBERT). Ngoài ra chúng ta chúng ta còn giúp mô hình *thích ứng* được với các từ mới ngoài 27 từ nằm ngoài từ điển của bộ dữ liệu, do đó mô hình được học sẽ tính *khái quát cao*.



Ảnh 4.7. Thay thế lớp Embedding của mạng LSTM bằng PhoBERT

Chúng ta thực hiện lớp thay thế lớp Embedding bằng **PhoBERT**, mỗi từ sẽ đi qua **PhoBERT** để *rút trích ra véc-tơ đặc trưng*, làm đầu vào cho đơn vị **LSTM** tại thời điểm t.



Ảnh 4.8. Mạng **LSTM** sử dụng **PhoBERT** cho biểu diễn

Việc mô hình thích ứng được với những từ mới đến từ phép biểu diễn *khái quát* của chúng trên không gian biểu diễn, những từ có ngữ nghĩa *tương đồng nhau* sẽ *gần nhau* trên không gian đó.

Chương 5. **THỬ NGHIỆM VÀ KẾT QUẢ**

5.1. Dữ liệu huấn luyện

Trong bài luận văn này, chúng tôi tiến hành nghiên cứu thực nghiệm dựa trên **3** tập dữ liệu sau:

- CSS (Color, Shape and Size)
- CSS-VN (phiên bản tiếng Việt của tập dữ liệu CSS)
- MIT-States

5.1.1. Mô tả tập dữ liệu

5.1.1.1. Tập dữ liệu CSS

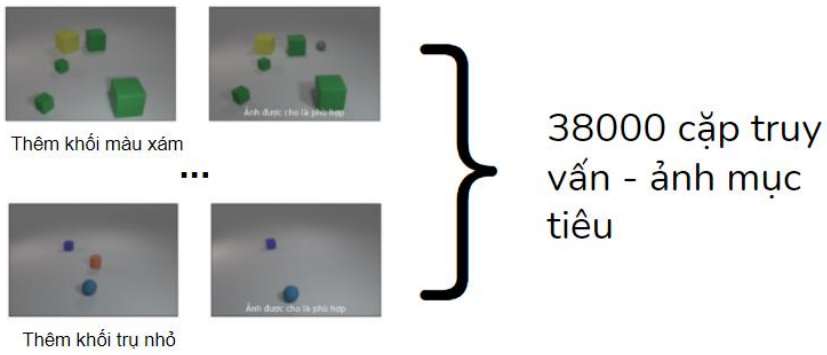
Tập dữ liệu **CSS** là tập dữ liệu được giới thiệu trong bài báo [1], được tạo ra để tập dữ liệu đánh giá cho bài toán truy vấn ảnh dựa trên ảnh và câu mô tả. Tập dữ liệu trên được xây dựng thông qua framework CLEVR, được sử dụng để tạo dữ liệu ảnh dựa trên đồ họa một cách tự động, để phục vụ cho các bài toán,

Bộ dữ liệu	**CSS**		
Tập chia	Huấn luyện	Phát triển	Kiểm định
Số lượng ảnh	19034	9518	9517
Số lượng câu truy vấn	18012	9029	9028
Số từ trong từ điển	27		

Bảng 5.1. Thống kê tập dữ liệu CSS

5.1.1.2. Tập dữ liệu CSS-VN

Tập dữ liệu **CSS-VN** là tập dữ liệu được chúng tôi xây dựng dựa trên tập dữ liệu CSS tiếng Anh được giới thiệu ở bài báo [1], bằng cách *thay thế* câu mô tả tăng cường bằng bản dịch của nó ở tiếng Việt.

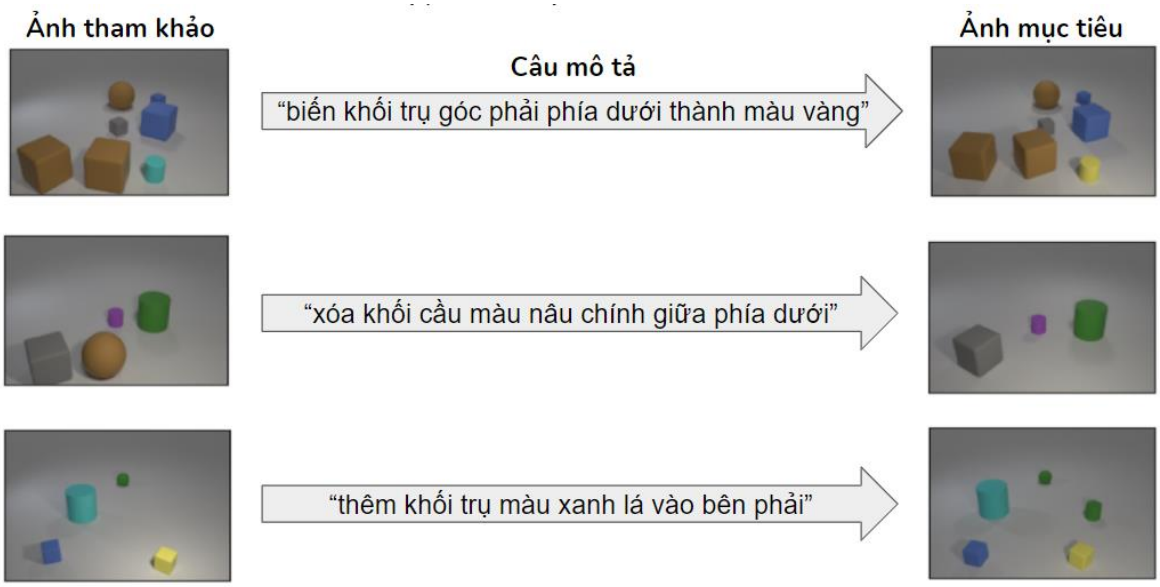


Ảnh 5.1. Minh họa cho tập dữ liệu **CSS-VN**

Tương tự với các tập dữ liệu đánh giá (benchmark) cho các bài toán truy vấn điển hình.

Bộ dữ liệu	**CSS-VN**		
Tập chia	Huấn luyện	Phát triển	Kiểm định
Số lượng ảnh	19034	9518	9517
Số lượng câu truy vấn	18012	9029	9028
Số từ trong từ điển	30		

Bảng 5.2. Thống kê bộ dữ liệu CSS-VN



Ảnh 5.21. Một số mẫu trong tập dữ liệu CSS-VN

Câu mô tả trong tập dữ liệu CSS-VN gồm:

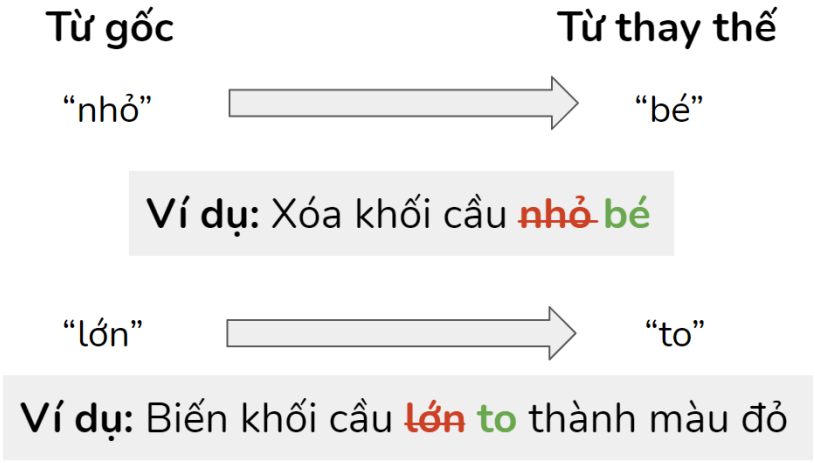
- **3 thao tác**: thêm, biến, xóa.
- **4 loại đối tượng**: khối, khối hộp chữ nhật, khối cầu, khối trụ.
- **8 màu**: màu nâu, màu xanh lá, màu xanh dương,...
- **9 vị trí**: bên trái, bên phải, trung tâm, góc phải phía dưới,..
- **2 kích thước**: lớn, bé

Quan sát tập dữ liệu ta sẽ thấy ở tiếng Việt tồn tại rất nhiều từ phân loại ("classifiers") như "khối", "màu" hay "bên"; sẽ rất tiềm năng khi ứng dụng lên các mô hình Học máy vì nó cho thêm thông tin về từ đứng sau nó.

5.1.1.3. Tập dữ liệu tăng cường CSS-VN-augmented

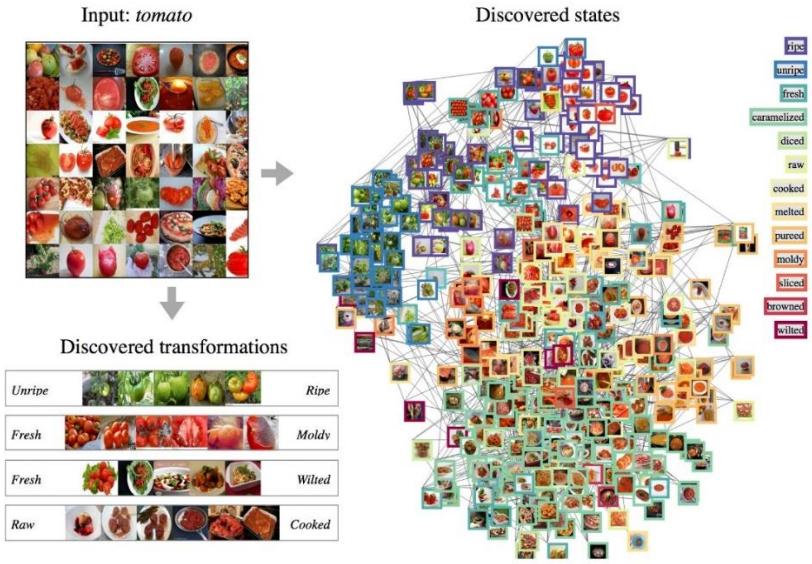
Tập dữ liệu **CSS-VN-augmented** là tập dữ liệu được xây dựng dựa trên tập dữ liệu **CSS-VN** với một số thay đổi về từ:

- "nhỏ" được thay thế thành "bé"
- "lớn" được thay thế thành "to"



Ảnh 5.3. Dữ liệu CSS-VN-augmented với những thay đổi nhỏ từ tập dữ liệu CSS-VN

5.1.1.4. Tập dữ liệu MIT-States:



Ảnh 5.4. Minh họa tập dữ liệu **MIT-States**

Tập dữ liệu **MIT-States** là một tập dữ liệu ảnh mô tả các biến đổi của các vật thể ở các trạng thái khác nhau. Ví dụ, đối với vật thể cà chua, tập dữ liệu có các mẫu ảnh từ trạng thái “tươi sống“ (fresh) cho tới “mốc meo“ (moldy)

Ảnh 5.5. Ảnh chuyển đổi trạng thái của cà chua từ "tươi sống" cho tới "mốc meo"

Theo tập dữ liệu này, mô hình sẽ nhận đầu vào là một câu truy vấn gồm:

- Ảnh ở trạng thái gốc
- Câu mô tả tăng cường về trạng thái đích

Đầu ra:

- Ảnh mục tiêu

Bộ dữ liệu	**MIT-States**		
Tập chia	Huấn luyện	Phát triển	Kiểm định
Số lượng ảnh	82732	9518	9517
Số lượng câu truy vấn	82732	5273	5273
Số từ trong từ điển	245		

Bảng 5.3. Thống kê bộ dữ liệu MIT-States

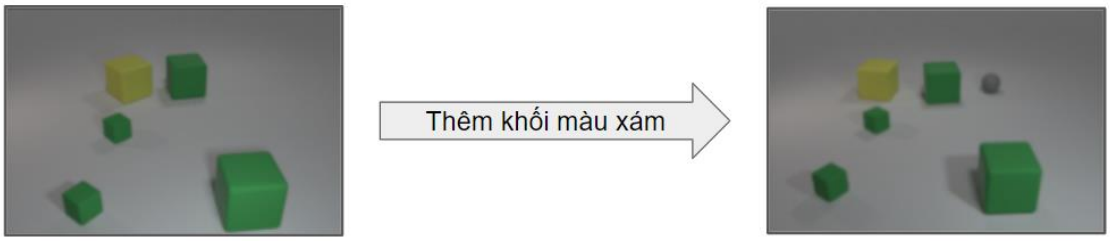
5.1.2. Động lực chọn tập dữ liệu

Khi lựa chọn tập dữ liệu huấn luyện, chúng tôi tham khảo các tập dữ liệu được sử dụng trong bài báo [1]. Đồng thời chúng tôi cũng chọn những bài báo để kiểm chứng các giả thuyết mà chúng tôi đặt ra có đúng hay không.

5.1.2.1. Động lực sử dụng 2 tập dữ liệu CSS và MIT-States

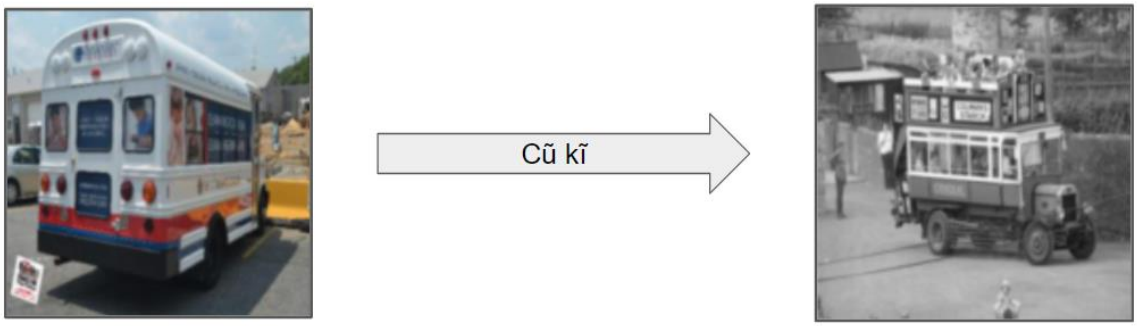
Trong bài nghiên cứu của chúng tôi, chúng tôi chọn sử dụng tới 2 tập dữ liệu để huấn luyện, đánh giá và báo cáo. Chúng tôi lựa chọn tập dữ liệu CSS và MIT-States cho nghiên cứu này do đây là hai tập dữ liệu này có đặc tính hoàn toàn khác nhau:

Đặc tính của tập dữ liệu CSS hay CSS-VN: Đây là tập dữ liệu bao gồm những khối hình hộp với cấu trúc hình học và màu sắc cơ bản, các biến đổi (bằng câu mô tả tăng cường) là các biến đổi cục bộ trên ảnh, mang các đặc tính về không gian. Ví dụ: "Thêm khối cầu", "Biến khối cầu thành màu xanh",..



Ảnh 5.6. Tập dữ liệu CSS với những biến đổi cục bộ

Đặc tính của tập dữ liệu MIT-States: Tập dữ liệu MIT-States là một tập dữ liệu gồm các , các phép biến đổi là các phép biến đổi toàn cục trên ảnh. Ví dụ: "Chuyển sang trạng thái tan chảy"



Ảnh 5.7. Tập dữ liệu MIT-States với những biến đổi toàn cục

Nhờ hai đặc tính khác nhau hoàn toàn của hai tập dữ liệu này, chúng sẽ phù hợp với 1 trong 2 mô hình khác nhau của TIRG được nêu ra trong bài báo, theo giả thuyết của tác giả đề ra. Chúng tôi muốn kiểm tra liệu giả thuyết này có thật sự đúng hay không.

5.1.2.2. Động lực xây dựng tập dữ liệu tiếng Việt CSS-VN

Mô hình được Text-Image Residual Gating cho bài toán truy vấn ảnh ban đầu được áp dụng trên tập dữ liệu TIRG gốc ở phiên bản tiếng Anh (với câu mô tả tăng cường là tiếng Anh). Chúng tôi muốn xây dựng tập dữ liệu tiếng Việt CSS-VN để có thể kiểm tra xem liệu mô hình trên có hoạt động tốt trên dữ liệu tiếng Việt hay không.

5.1.2.3. Động lực xây dựng tập dữ liệu tăng cường CSS-VN-augmented

Sau khi hiện thực hóa lại mô hình TIRG, chúng tôi tiến hành cải tiến mô hình TIRG, cụ thể là thay thế lớp embedding trong mô-đun biểu diễn ảnh LSTM thành một mô hình ngôn ngữ được tiền huấn luyện là PhoBERT. Mục tiêu của việc cải tiến này là để mô hình TIRG có thể thích ứng được với những từ nằm ngoài từ điển của tập huấn luyện, lý do là tập huấn luyện CSS-VN có số lượng từ vựng rất hạn chế (chỉ 27 từ).

Do đó để đánh giá tính thích ứng của mô hình với những dữ liệu có câu mô tả tăng cường chứa từ nằm ngoài từ điển của dữ liệu huấn luyện, chúng tôi xây dựng tập dữ liệu CSS-VN-augmented với những thay đổi tuy nhỏ nhưng đủ để đánh giá mô hình mới.

5.2. Thang đo đánh giá

Để đánh giá hệ thống truy vấn, chúng tôi sử dụng độ **Recall at rank K** ở 5 mức thứ hạng như sau: 1, 5, 10, 50 và 100. Recall@k là một trong những phép đo dựa trên thứ hạng (ranking-based metrics) của hệ thống truy vấn.

Khi sử dụng phép đo Recall-at-k, ta cho rằng một hệ thống tìm kiếm tốt là một hệ thống có thể trả về tất cả các kết quả phù hợp ở các vị trí đầu tiên. Recall@k có công thức như sau:

$$R@k = \frac{1}{|U|}\sum_{u \in U}\frac{|Rel_u@k|}{|Rel_u|}$$

Trong đó,

- $Rel_u@k$ là tất cả kết quả được cho là phù hợp trong số các kết quả đầu tiên được trả về của truy vấn u cho tới kết quả thứ k
- Rel_u là tất cả kết quả được cho là phù hợp trong các kết quả được trả về của truy vấn u
- U là tập các hợp các câu truy vấn

5.3. Kết quả

5.3.1. Cấu hình huấn luyện

Trong xuyên suốt quá trình thực nghiệm, chúng tôi sử dụng cấu hình sau cho tất cả các phiên huấn luyện trên tập dữ liệu CSS và CSS-VN:

Cấu hình	Tham khảo	Sử dụng
Tốc độ học (LR)	0.01	0.01
Số lần lặp	160,000	594,000
Kích thước batch	32	32

Bảng 5.4. Cấu hình huấn luyện trên tập dữ liệu CSS và CSS-VN

Đối với tập dữ liệu MIT-States, chúng tôi sử dụng cấu hình sau:

Cấu hình	Tham khảo	Sử dụng
Tốc độ học (LR)	0.01	0.01
LR decay	5e-6	5e-6
Chu kì decay	50,000	50,000
Số lần lặp	160,000	594,000
Kích thước batch	32	32

Bảng 5.5. Cấu hình huấn luyện trên tập dữ liệu MIT-States

Cấu hình trên được chúng tôi tham khảo từ bài báo gốc, với thay đổi nhỏ từ 160,000 lần lặp lên thành 594,000 để có thể huấn luyện mô hình dài hơi hơn.

5.3.2. Tái hiện mô hình TIRG trên dữ liệu tiếng Anh

Ở thí nghiệm này, chúng tôi tái hiện lại kết quả của bài báo, để kiểm tra xem kết quả có tương đồng với kết quả bài báo hay không để chúng tôi có thể sử dụng mô hình tái hiện làm mô hình cơ sở (hay *baseline*) để phát triển lên tiếp. Đồng thời chúng tôi cũng rút ra một số kết luận và góc nhìn về kết quả được tái hiện.

5.3.2.1. Tái hiện trên tập dữ liệu CSS

	Cấu hình	R@1	R@5	R@10	R@50	R@100
Bài báo công bố	TIRG-Conv	**73.7**	KCB*	KCB	KCB	KCB
	TIRG-FC	71.2	KCB	KCB	KCB	KCB
Kết quả tái hiện	TIRG-Conv	**71.11**	90.62	94.13	98.19	99.09
	TIRG-FC	70.77	**91.14**	**94.54**	**98.43**	**99.17**

Bảng 5.6. Kết quả tái hiện trên tập dữ liệu CSS (KCB*: không công bố)

5.3.2.2. Tái hiện trên tập dữ liệu MIT-States

	Cấu hình	R@1	R@5	R@10	R@50	R@100
Bài báo công bố	TIRG-Conv	10.3	KCB*	KCB	KCB	KCB
	TIRG-FC	**12.2**	31.9	43.1	KCB	KCB
Kết quả tái hiện	TIRG-Conv	10.49	29.74	40.69	68.59	**96.08**
	TIRG-FC	**13.25**	**32.51**	**43.65**	**69.72**	95.93

Bảng 5.7. Kết quả tái hiện trên tập dữ liệu MIT-States (KCB*: không công bố)

5.3.2.3. Nhận xét

Kết quả tái hiện cho ra kết quả ở độ đo R@1 khá tương đồng với kết quả của bài báo công bố. Mô hình TIRG-Conv sẽ cho ra kết quả tốt hơn trên tập dữ liệu CSS hơn MIT do tính chất biến đổi cục bộ của bộ dữ liệu CSS. Ngược lại, mô hình TIRG-FC cho ra kết quả vượt trội hơn trên tập dữ liệu MIT-States, do tính chất biến đổi toàn cục của tập dữ liệu MIT-States.

Một điều đáng chú ý là khi huấn luyện trên tập dữ liệu CSS, TIRG-Conv chỉ ưu việt hơn TIRG-FC trên độ đo R@1, còn lại kém hơn TIRG-FC ở các độ đo còn lại R@5, R@10, R@50 và R@100.

5.3.3. Mô hình TIRG trên dữ liệu CSS-VN

Trong thí nghiệm này, chúng tôi thực hiện việc huấn luyện hai mô hình TIRG-Conv và mô hình TIRG-FC trên tập dữ liệu CSS-VN do chúng tôi xây dựng. Mục tiêu là để tìm hiểu liệu mô hình này có thích hợp đối với các câu mô tả tiếng Việt không. Đồng thời đúc kết được một số góc nhìn sâu sắc và chia sẻ những khám phá trong lúc huấn luyện các mô hình trên.

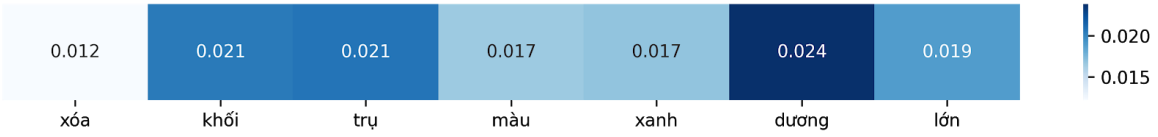
Cấu hình	R@1	R@5	R@10	R@50	R@100
TIRG-FC	**78.03**	**94.91**	**97.18**	**98.28**	**99.57**
TIRG-Conv	75.29	92.6	95.48	98.69	99.29

Bảng 5.8. Kết quả thực nghiệm TIRG-FC và TIRG-Conv trên tập CSS-VN

Khi được huấn luyện trên dữ liệu tiếng Việt **CSS-VN**, mô hình **TIRG** cho ra kết quả khả quan, với R@1 là **78.03,** cao hơn nhiều so với tập dữ liệu CSS bản tiếng Anh. Để hiểu hơn về bản chất của việc huấn luyện **LSTM** trên tập dữ liệu tiếng Việt, chúng tôi tiến hành *trực quan hóa tính quan trọng* của các từ trong việc mô hình hóa chuỗi câu trong **LSTM**. Để đo cho tính quan trọng của một từ t_i trong một câu chúng tôi thiết lập một phép đo **L2** giữa biểu diễn từ đó ở thời điểm i với thời điểm trước đó là $(i-1)$. Với $h_t \in R^{(-1,1)}$

$$significant(t_i) = \|h_i - h_{i-1}\|_2$$

Một từ được cho là quan trọng khi khoảng cách *trạng thái ẩn* ở thời điểm của từ đó so với thời điểm trước đó là lớn. Điều này có nghĩa có nhiều "thông tin" được cập nhật đối với từ này so với thời điểm trước đó.



Ảnh 5.8. Trực quan hóa LSTM 1

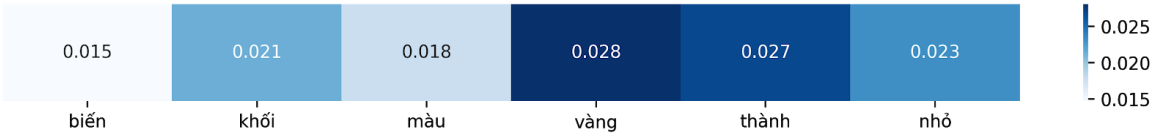
LSTM sẽ tập trung cập nhật "thông tin" cho những từ có *hàm lượng thông tin cao*, ví dụ như là từ "*trụ*", "*dương*" hay "*lớn*". Những từ khác trả lời các câu hỏi "*xóa khối gì?*", "*ở đâu?*", "*màu gì?*" và "*đặc điểm ra sao?*"

Cùng là những từ biểu diễn màu sắc như từ "*dương*" nhưng những từ khác như "*màu*" và "*xanh*" được cho là những từ kém quan trọng hơn vì tồn tại các biến thể khác của màu xanh như "màu xanh lá". Do đó từ "dương" có tính định danh cao.

Ảnh 5.9. Trực quan hóa LSTM 2

Một ví dụ khác cho thấy mô hình **LSTM** cũng học được cách cập nhật thông tin thông qua mỗi thời điểm, qua đó có thể thấy những từ mang hàm lượng thông

tin nhiều bao gồm từ "khối", "dương", "chính", "trên", "thành" và "lá". Đây là những từ cực quan trọng và mang tính định danh cao khi mô hình hóa các câu mô tả, cũng rất tương đồng với cách con người tiếp nhận các câu trên.



Ảnh 5.10. Trực quan hóa LSTM 3

LSTM đã thành công trong việc mô hình hóa câu mô tả tăng cường với dữ liệu tiếng Việt do sử dụng nhiều từ phân loại, mà cụ thể trong tập dữ liệu của ta là từ "khối". Với cùng mô hình **TIRG-FC**, chúng ta sẽ cũng so sánh **R@1** của mô hình TIRG đối với các đối tượng *"object", "cube", "sphere",* và *"cylinder"* khi được huấn luyện trên dữ liệu **CSS** và các đối tượng tượng ứng của nó *ở* **CSS-VN**.

	CSS	CSS-VN
"object"/"khối"	81.03%	**82.64%**
"cube"/"khối lập phương"	53.99%	**65.13%**
"sphere"/"khối cầu"	80.69%	**83.44%**
"cylinder"/"khối trụ"	57.42%	**73.03%**

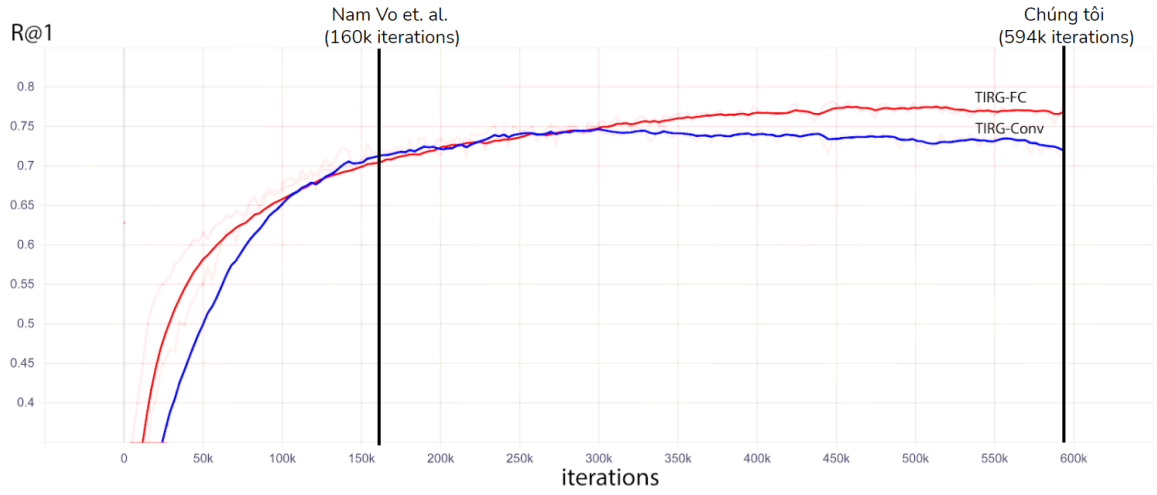
Có thể thấy, tất cả câu chứa các đối tượng tương ứng của tập CSS ở tập CSS-VN khi đánh giá trên mô hình R@1 đều ra chỉ số cao hơn CSS tiếng Anh. Việc xuất hiện thêm từ "khối" khi dịch từ Anh sang Việt cho mô hình thêm nhiều thông tin và mối quan hệ giữa các đối tượng trên khi cung cấp thông tin cho từ đứng sau nó. Điều

Ngoài ra chúng tôi quan sát thấy kết quả chưa giống với giả thuyết mà tác giả đề ra về tính cục bộ và toàn cục của dữ liệu. Trong đó, và một biến thể thực hiện việc kết hợp ở lớp fully connected (FC), gọi là **TIRG-FC**; Một biến thể khác thực hiện phép kết hợp ở lớp convolution cuối, gọi là **TIRG-Conv**.

Theo tác giả:

- Đối với tập **CSS-VN**, những biến đổi (dựa trên câu mô tả tăng cường) là những *biến đổi cục bộ* trên không gian ảnh (spatially localized), cụ thể là "thêm", "xóa" hoặc "biến" các đối tượng bên trong ảnh, sẽ phù hợp hơn với việc thao tác trên bản đồ đặc trưng của lớp Convolution do bản đồ đặc trưng mang thông tin cục bộ trong bức ảnh, sẽ phù hợp với **TIRG-Conv** hơn.
- Đối với tập **MIT-States**, những biến đổi này là những *biến đổi toàn cục* (global) trên toàn bộ bức ảnh, do đó việc thao tác trên véc-tơ đặc trưng sẽ phù hợp hơn vì véc-tơ đặc trưng chứa đựng nhiều thông tin toàn cục của bức ảnh, sẽ phù hợp với **TIRG-FC** hơn.

Tuy vậy khi thực nghiệm với cấu hình của chúng tôi với TIRG-FC cho ra kết quả tốt hơn TIRG-FC trên tập CSS-VN khi huấn luyện với cấu hình của chúng tôi. Kết quả của tác giả có thể giải thích được do tác giả huấn luyện với số iterations ít hơn chúng tôi.



Ảnh 5.11. R@1 của TIRG-FC và TIRG-CONV khi huấn luyện trên tập dữ liệu CSS-VN

Với 160,000 lần lặp như tác giả đề **CSS-VN** xuất, **TIRG-Conv** sẽ cho ra kết quả tốt hơn trên tập. Tuy nhiên từ lần lặp thứ 300,000 trở đi, **TIRG-FC** cho ra kết quả tốt

và vượt trội hơn nhiều **TIRG-Conv.** Lưu ý rằng giá trị R@1 của mỗi mô hình được chúng tôi lấy kết quả R@1 tốt nhất ở mọi lần lặp.

5.3.4. Cải tiến mô hình TIRG với PhoBERT

Trong thí nghiệm này, chúng tôi thực hiện thay thế lớp Embedding bằng mô hình ngôn ngữ được tiền huấn luyện là **PhoBERT** để tăng độ thích ứng của mô hình đối với những từ mới nằm ngoài trong từ điển của câu mô tả tăng cường trong tập huấn luyện, đồng thời có thể giảm thời gian huấn luyện của mô hình trên tập do không phải huấn luyện lại lớp Embedding.

	R@1	R@5	R@10	R@50	R@100
TIRG-FC-Embedding	78.03	94.91	97.18	98.28	99.57
TIRG-FC-PhoBERT	**78.05**	**96.30**	**97.91**	**99.44**	**99.66**

Bảng 5.9. Kết quả truy vấn của TIRG-FC-Embedding và TIRG-FC-PhoBERT trên các mức Recall khác nhau

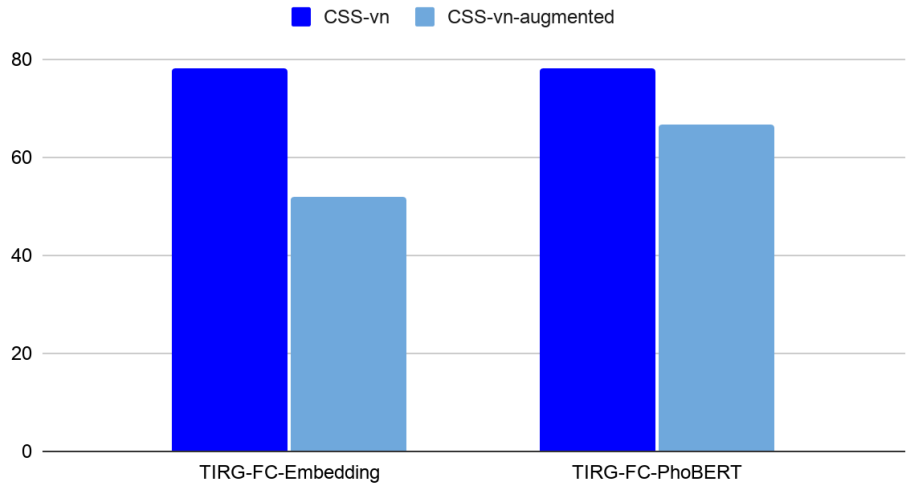
TIRG-FC-PhoBERT cho kết quả tốt hơn **TIRG-FC-Embedding** trên tập dữ liệu **CSS-VN**, song kết quả rất tương đồng ở **R@1.** Điểm khác nhau duy nhất là mô hình **TIRG-FC-PhoBERT** được tin rằng sẽ *khái quát hóa* tốt hơn trên những tập dữ liệu chứa những từ chưa được học.

Do đó đó chúng tôi thực hiện việc so sánh chỉ số R@1 hai mô hình trên tập dữ liệu tăng cường **CSS-VN-augmented** đối vs tập dữ liệu gốc CSS-VN.

	CSS-VN (a)	CSS-VN-augmented (b)	(a) – (b)
TIRG-FC-Embedding	0.7803	0.5193	0.261
TIRG-FC-PhoBERT	0.7805	**0.6651**	**0.1154**

Bảng 5.10. So sánh TIRG-FC-Embedding và TIRG-FC-PhoBERT trên CSS-VN và CSS-VN-augmented

Mô hình **TIRG-FC-Embedding** và **TIRG-FC-PhoBERT** cho kết quả *tương đồng* khi đánh giá trên **CSS-VN**, tuy nhiên khi đánh giá trên tập **CSS-VN-augmented** thì **TIRG-FC-Embedding** bị *giảm đáng kể*.



Ảnh 5.12. So sánh độ thích ứng của TIRG-Embedding và TIRG-PhoBERT

Có thể thấy, **TIRG-FC-PhoBERT** thích ứng tốt hơn trên tập dữ liệu tăng cường với một số thay đổi nhỏ về ngôn ngữ ở câu mô tả tăng cường. Với phép biểu diễn từ của **PhoBERT**, vốn được tiền huấn luyện trên một tập ngữ liệu lớn gồm 20GB đoạn văn bản, sẽ có khả năng biểu diễn khái quát hơn việc chỉ sử dụng một lớp **Embedding** vốn hoạt động tốt với các từ nằm trong từ điển.

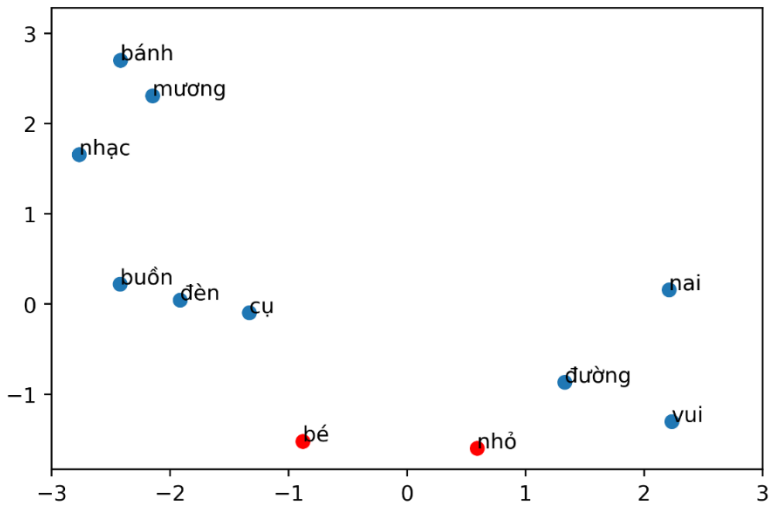
Bên dưới chúng tôi thống **R@1** đối với những mẩu chứa câu mô tả có từ *"bé"* và "to".

	"bé"	**"to"**
TIRG-FC-Embedding	0.4258	0.4306
TIRG-FC-PhoBERT	**0.7906**	**0.7665**

Bảng 5.11. So sánh R@1 của **TIRG-FC-Embedding và TIRG-FC-PhoBERT** trên những câu chứa từ thay thế là *"bé"* và *"to"*

Mô hình **TIRG-PhoBERT** thích ứng tốt hơn đối với các mẫu dữ liệu biến đổi, cho ra kết quả R@1 tốt hơn 30% trở lên.

Điều này có thể giải thích vì biểu diễn của "nhỏ" đối với "bé" và "to" đối với "lớn" trên không gian biểu diễn của **PhoBERT** là gần nhau.



Ảnh 5.13. Trực quan hóa biểu diễn từ trên không gian 2D

Trong hình trên chúng tôi tôi trực quan hóa biểu diễn từ của **PhoBERT** bằng biểu đồ. Có thể thấy những từ có đặc điểm về mặc *ngữ nghĩa tương đồng* thì sẽ nằm gần nhau trên không gian biểu diễn của **PhoBERT**, ví dụ: *"nhỏ"* sẽ gần với *"bé"*, *"lớn"* sẽ gần với *"to"*. Nếu sử dụng lớp Embedding thông thường, những từ mới chưa được học sẽ được ánh xạ thành vectơ bất kì, thường là vectơ $\vec{0}$.

Việc trực quan hóa trên không gian nhiều chiều (768 chiều của PhoBERT) không khả thi, nên chúng tôi sử dụng phương pháp **Principal Component Analysis** (PCA) để giảm số chiều xuống còn 2 để tiện cho việc vẽ biểu đồ.

5.3.5. Nghiên cứu cắt bỏ

Ở trong thí nghiệm cắt bỏ này, chúng tôi muốn kiểm tra tính hiệu quả của mô-đun kết hợp **TIRG** bằng cách thay thế bằng **Concatenation**, vốn là một cách kết hợp đơn giản được sử dụng trong rất nhiều ứng dụng [10, 11, 12,13]. Ngoài ra chúng tôi còn thử chỉ sử dụng ảnh hoặc câu mô tả cho truy vấn. Thí nghiệm này được thực hiện trên tập dữ liệu **CSS-VN**.

	R@1	R@5	R@10	R@50	R@100
Chỉ dùng ảnh	06.59	29.04	53.08	94.09	96.51
Chỉ dùng câu mô tả	0.171	0.504	0.863	2.31	3.611
Concatenation	69.09	90.00	93.83	98.22	99.00
TIRG-FC	**78.03**	**94.91**	**97.18**	**98.28**	**99.57**
TIRG-Conv	75.29	92.6	95.48	98.69	99.29

Bảng 5.12. Nghiên cứu cắt bỏ về các mô-đun kết hợp ảnh và văn bản

Khi huấn luyện chỉ dùng ảnh hoặc câu mô tả, kết quả rất tệ với **R@1** tương ứng là *6.59%* và *0.161%.* Phương pháp **Concatenation** cho ra kết quả với **R@1** là *69.09%,* kém hơn **TIRG-FC** đến 8.94%. Phương pháp **TIRG-FC** và **TIRG-Conv** cho kết quả tốt hơn 3 phương pháp còn lại với độ đo **R@1** lần lượt là *78.03%* và *75.29%.*

Có thể thấy, mô-đun kết hợp **TIRG** là một mô-đun kết hợp hiệu quả trong việc kết hợp đặc trưng ảnh và đặc trưng văn bản cho truy vấn ảnh, song chúng tôi vẫn chưa có thời gian để thử nghiệm trên các

Bên cạnh đó, chúng tôi còn tiến hành cắt bỏ bộ tách từ **RDRSegmenter** ra khỏi mô hình **TIRG-FC**.

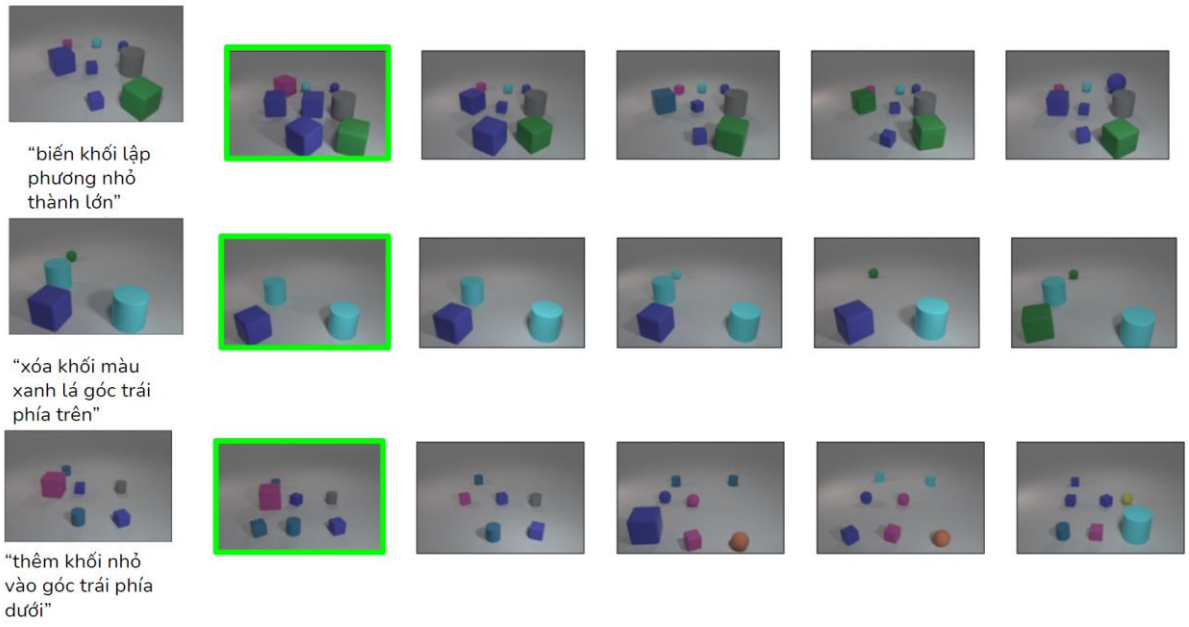
Cấu hình	R@1	R@5	R@10	R@50	R@100
Sử dụng RDRSegmenter	**78.03**	**94.91**	**97.18**	98.28	**99.57**
Không RDRSegmenter	74.45	92.83	95.61	**98.72**	99.22

Bảng 5.13. Nghiên cứu cắt bỏ trên mô-đun tách từ

Bộ tách từ **RDRSegmenter** đóng một vai trò quan trọng trong việc cải thiện kết quả mô hình hóa do lược bỏ sự *nhặp nhằng khoảng trắng* ở tiếng Việt. Sau khi tách từ, từ "lập phương" và "trung tâm" được token chung lại với nhau tương ứng thành "chữ_nhật" và "trung_tâm", do đó kết quả **R@1** lại cải thiện rất đăng kể.

5.3.6. Một số kết quả truy vấn mẫu

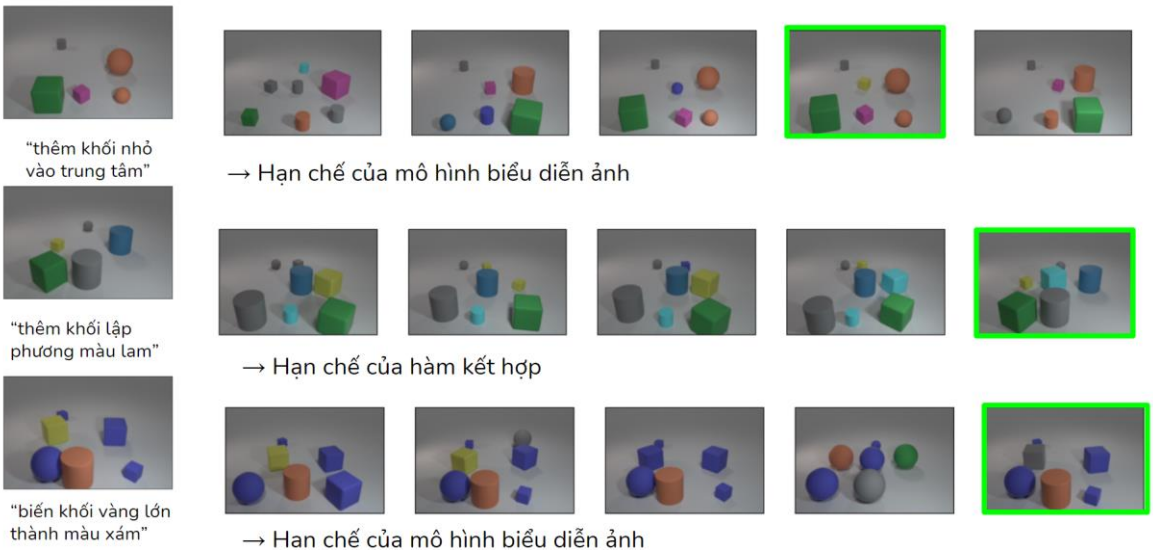
Ở phần này, chúng tôi sẽ trình bày một số kết quả truy vấn mẫu của mô hình được huấn luyện và đưa ra một số nhận xét. Bên dưới là một số kết quả tốt, trong đó tất cả tấm ảnh phù hợp trả về ở vị trí thứ hạng đầu tiên.



Ảnh 5.14. Kết quả truy vấn mẫu 1

Có thể thấy, ở những mẫu này, hệ thống đã trả về được kết quả phù hợp ở vị trí thứ nhất. Những kết quả khác ở các mức thứ hạng tiếp theo cũng cho những tấm ảnh rất tương đồng về mặt thị giác.

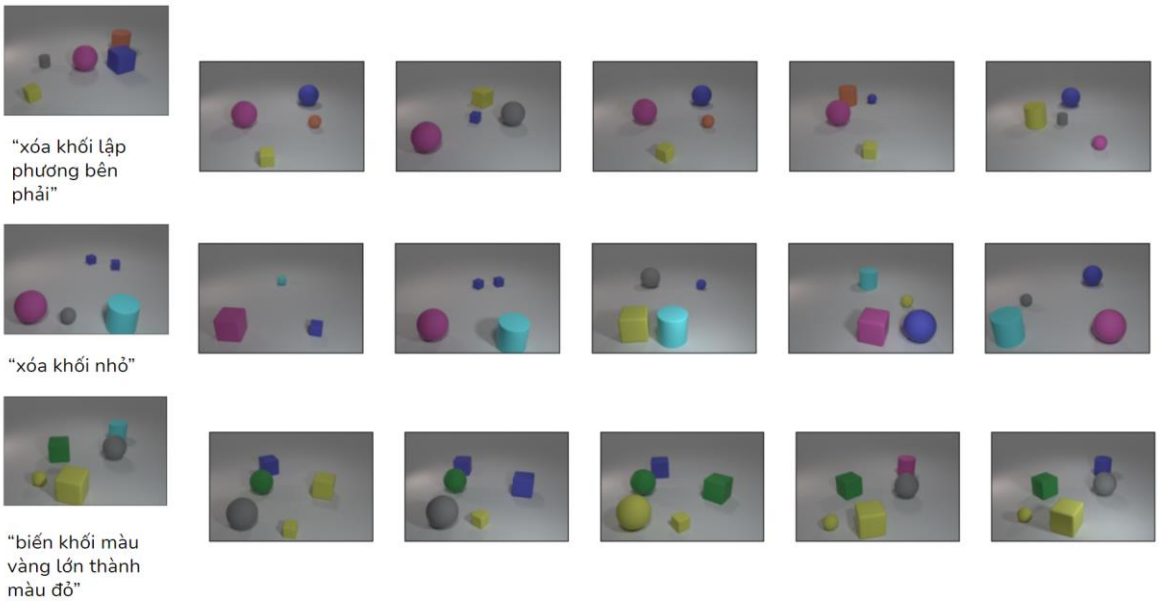
Song ở các kết quả tiếp theo, mô hình đã không cho ra được các kết quả phù hợp ở vị trí đầu.



Ảnh 5.15. Kết quả truy vấn mẫu 2

Có thể thấy ở truy vấn thứ nhất và truy vấn thứ 3, kết quả không những không trả về ở vị trí đầu, tuy nhiên các thứ hạng kế đó cũng có thể nhận ra chúng không tương đồng về mặt thị giác, điều này đến từ hạn chế của mô hình biểu diễn ảnh. Ở truy vấn thứ 2, tuy những kết quả trả về không rơi vào vị trí đầu, đây là một hạn chế của hàm $f_{kết\ hợp}$.

Ngoài ra, chúng ta có những mẩu truy vấn không trả về được kết quả ở 5 thứ hạng đầu tiên.



Ảnh 5.16. Kết quả truy vấn mẫu 3

Có thể thấy, ở những mẩu này, kết quả không những không được trả về ở 5 mức thứ hạng đầu tiên, mà những kết quả truy vấn ra cũng không có nhiều đặc điểm tương đồng với tấm ảnh đích. Đây là động lực để chúng ta thiết kế lại bộ biểu diễn ảnh và hàm $f_{kết\ hợp}$.

Chương 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. Kết luận

Thông qua quá trình nghiên cứu, chúng tôi đã rút ra một số kết luận sau:

- Nhu cầu truy vấn ảnh sử dụng kết hợp ảnh và câu mô tả tăng cường sẽ trở thành một xu hướng mới trong tương lai khi nó giải quyết vấn đề khoảng cách ý định của người dùng, cung cấp một giao thức cho phép người dùng linh hoạt mô tả, chia sẻ nhu cầu thông tin của mình vào hệ thống tìm kiếm một cách tường minh hơn do có thể sử dụng một tấm ảnh tham khảo đã có sẵn kèm với một số thay đổi mong muốn dưới dạng văn bản mô tả.
- Mô hình **TIRG** là một mô hình hiệu quả trong việc kết hợp đặc trưng ảnh và văn bản. Việc này thể hiện qua nghiên cứu cắt bỏ phép kết hợp TIRG bằng cách thay thế bằng các phương pháp khác như Concatenation, FiLM, hoặc chỉ sử dụng ảnh hay câu mô tả để truy vấn.
- Mô hình **TIRG** hoạt động tốt trên tập dữ liệu tiếng Việt do chúng tôi xây dựng nên nhờ những ưu điểm của tiếng Việt trên các mô hình Học máy là sở hữu lượng từ phân loại giàu có và không có biến tố. Qua quá trình trực quan hóa **LSTM** chúng ta cũng đã thấy được cách **LSTM** biểu diễn văn bản qua từng thời điểm rất tương đồng với cách con người khi tiếp cận với các câu mô tả.
- Việc thay thế lớp **Embedding** trong **LSTM** của mô-đun **TIRG** bằng **PhoBERT** đã giúp mô hình tăng độ thích ứng với những từ mới không có mặt trong từ do mô hình **PhoBERT** là mô hình ngôn ngữ được tiền huấn luyện với một lượng dữ liệu khổng lồ. Nhờ PhoBERT chúng tôi có thể giảm thời gian huấn luyện đi rất nhiều nhờ không phải huấn luyện lớp biểu diễn trong tổng thể mạng TIRG.

- Việc sử dụng **RDRSegmenter** làm bộ tách từ tăng hiệu quả khi truy vấn do giải quyết nhập nhằng khoảng trắng, nhờ vậy mà mô hình hóa tiếng Việt được hiệu quả hơn.
- Công cụ dịch dựa trên tập luật **URBANS** đã cho phép chúng tôi xây dựng bộ dữ liệu tiếng Việt **CSS-VN** với *nỗ lực tối thiểu*, với *ít công sức* và *hoàn toàn tự động*. Với công cụ dịch trên, chúng tôi đã tạo ra nhiều tập dữ liệu khác nhau với các tập luật khác nhau một cách nhanh chóng, từ đó có thể xúc tiến việc thử nghiệm một cách *linh hoạt và hiệu quả* hơn.

6.2. Hướng phát triển

Sau thời gian thực hiện khóa luận tốt nghiệp, chúng tôi cũng nung nấu cho mình một số ý tưởng:

- **Truy vấn ảnh dựa trên đối thoại.** Mô hình TIRG có thể phát triển để giải quyết bài toán truy vấn ảnh dựa trên đối thoại [29], trong đó câu mô tả được người dùng cung cấp liên tục để củng cố hệ thống tìm kiếm về nhu cầu thông tin. Khi này, phép biểu diễn kết hợp ϕ_{xt} sẽ liên tục được cập nhật với câu mô tả mới $\phi_{t'}$.
- **Áp dụng phương pháp kết hợp TIRG cho bài toán khác.** Phương pháp TIRG là một phương pháp hiệu quả trong biểu diễn cặp ảnh và văn bản. Do đó, phương pháp này rất tiềm năng để áp dụng các bài toán như Hỏi đáp trên ảnh (visual question answering) [10] hay chỉnh sửa ảnh đối thoại (conversation image editing) [28], một bài toán mới được định nghĩa gần đây.
- **Cải tiến các mô-đun biểu diễn trong TIRG.** TIRG vốn được cấu thành bởi hai bộ biểu diễn là ảnh và văn bản. Việc cải tiến khả dĩ có thể được thực hiện trên việc cải tiến trên hai mô-đun này. Một hướng cải tiến là thay thế LSTM bằng một mô hình dựa trên Attention như BERT [34] hay Alberta. Đồng thời, chúng ta có thể cải tiến mô-đun biểu diễn ảnh thành các mạng biểu diễn ảnh state-of-the-art hiện nay như EfficientNet [61].

- **Cập nhật công cụ dịch để hỗ trợ phân tích cú pháp với ràng buộc ngữ nghĩa**. Hiện tại hướng dịch của chúng tôi dựa chủ yếu trên việc phân tích cây cú pháp phi ngữ cảnh (context-free grammar). Việc cho phép phân tích cây cú pháp dựa trên ngữ cảnh sẽ giúp quá trình dịch trở nên linh hoạt hơn do có thêm ràng buộc về ngữ nghĩa các từ trước khi thực hiện quá trình biến đổi câu cú pháp và biến đổi một-một giữa 2 ngôn ngữ.

TÀI LIỆU THAM KHẢO

[1] Vo, Nam, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. "Composing text and image for image retrieval-an empirical odyssey." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6439-6448. 2019.

[2] Hoffer, Elad, and Nir Ailon. "Deep metric learning using triplet network." In *International Workshop on Similarity-Based Pattern Recognition*, pp. 84-92. Springer, Cham, 2015.

[3] Koch, Gregory, Richard Zemel, and Ruslan Salakhutdinov. "Siamese neural networks for one-shot image recognition." In *ICML deep learning workshop*, vol. 2. 2015.

[4] Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815-823. 2015.

[5] Truong-Phat Nguyen. "URBANS: Universal Rule-Based Machine Translation toolkit." https://github.com/pyurbans/urbans, 2021.

[6] N. Kanopoulos, N. Vasanthavada and R. L. Baker, "Design of an image edge detection filter using the Sobel operator," in *IEEE Journal of Solid-State Circuits*, vol. 23, no. 2, pp. 358-367, April 1988, doi: 10.1109/4.996.

[7] J. Canny, "A Computational Approach to Edge Detection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679-698, Nov. 1986, doi: 10.1109/TPAMI.1986.4767851.

[8] Lowe, David G.. "Distinctive Image Features from Scale-Invariant Keypoints".*Int. J. Comput. Vision* 60, no.2, 2004: 91–110.

[9] Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. *Learning internal representations by error propagation*. No. ICS-8506. California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[10] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In ICCV, 2015.

[11] X. Guo, H. Wu, Y. Cheng, S. Rennie, and R. S. Feris. Dialog-based interactive image retrieval. arXiv preprint arXiv:1805.00145, 2018.

[12] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737, 2017.

[13] I. Misra, A. Gupta, and M. Hebert. From red wine to red tomato: Composition with context. In CVPR, 2017.

[14] Hoffman DD, Richards WA. Parts of recognition. Cognition. 1984 Dec 1;18(1-3):65-96.

[15] Zhu, Song-Chun, and David Mumford. *A stochastic grammar of images*. Now Publishers Inc, 2007.

[16] İkizler, N. and Forsyth, D.A., 2008. Searching for complex human activities with no visual examples. *International Journal of Computer Vision*, *80*(3), pp.337-357.

[17] Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D. Object detection with discriminatively trained part-based models. IEEE transactions on pattern analysis and machine intelligence. 2009 Sep 22;32(9):1627-45.

[18] Andreas J, Rohrbach M, Darrell T, Klein D. Neural module networks. InProceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 39-48).

[19] Johnson J, Hariharan B, Van Der Maaten L, Hoffman J, Fei-Fei L, Lawrence Zitnick C, Girshick R. Inferring and executing programs for visual reasoning.

InProceedings of the IEEE International Conference on Computer Vision 2017 (pp. 2989-2998).

[20] Kato K, Li Y, Gupta A. Compositional learning for human object interaction. InProceedings of the European Conference on Computer Vision (ECCV) 2018 (pp. 234-251).

[21] Miyato T, Koyama M. cGANs with projection discriminator. arXiv preprint arXiv:1802.05637. 2018 Feb 15.

[22] Nguyen DQ, Nguyen AT. PhoBERT: Pre-trained language models for Vietnamese. arXiv preprint arXiv:2003.00744. 2020 Mar 2.

[23] Bishop CM. Pattern recognition and machine learning. springer; 2006.

[24] Manning CD, Schütze H, Raghavan P. Introduction to information retrieval. Cambridge university press; 2008.

[25] N. N. Vo and J. Hays. Localizing and orienting street views using overhead imagery. In ECCV, 2016.

[26] LeCun, Yann, and Yoshua Bengio. "Convolutional networks for images, speech, and time series." *The handbook of brain theory and neural networks* 3361, no. 10 (1995): 1995.

[27] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016.

[28] Manuvinakurike, Kallirroi. "Conversational Image Editing: Incremental Intent Identification in a New Dialogue Task." . In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue* (pp. 284–295). Association for Computational Linguistics, 2018.

[29] Xiaoxiao Guo*, Hui Wu*, Yu Cheng, Steven Rennie, Gerald Tesauro, Rogerio Schmidt Feris. “Dialog-based Interactive Image Retrieval.” NeurIPS, 2018.

[30] Vinyals, Oriol, Alexander Toshev, Samy Bengio, and Dumitru Erhan. "Show and tell: A neural image caption generator." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156-3164. 2015.

[31] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer. 2018.

[32] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In NIPS, 2017.

[33] H. Noh, P. Hongsuck Seo, and B. Han. Image question answering using convolutional neural network with dynamic parameter prediction. In CVPR, 2016.

[34] Jacob Devlin, , Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.", 2019.

[35] Zhenzhong Lan, , Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations.", 2020.

[36] Medium. 2021. *An Intuitive Explanation Of Gradient Descent*. [online] Available at: <https://towardsdatascience.com/an-intuitive-explanation-of-gradient-descent-83adf68c9c33> [Accessed 20 January 2021].

[37] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[38] Krizhevsky, Alex, Ilya, Sutskever, and Geoffrey E, Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." . In *Advances in Neural Information Processing Systems* (pp. 1097–1105). Curran Associates, Inc., 2012.

[39] Karen Simonyan, , and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition.", 2015.

[40] Christian Szegedy, , Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going Deeper with Convolutions.", 2014.

[41] Kaiming He, , Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition.", 2015.

[42] Cotterell, Ryan, Sabrina J. Mielke, Jason Eisner, and Brian Roark. "Are all languages equally hard to language-model?." *arXiv preprint arXiv:1806.03743*, 2018.

[43] Đình-Hòa Nguyễn, "Tiếng Việt không son phấn", *John Benjamins Publishing Company*, 1997.

[44] Dat Quoc Nguyen, , Dai Quoc Nguyen, Thanh Vu, Mark Dras, and Mark Johnson. "A Fast and Accurate Vietnamese Word Segmenter." . In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)* (pp. 2582–2587).2018.

[45] Abi Anvita, "Reduplication in South Asian languages", *New Delhi: Allied Publishers.*, 1992.

[46] Tran, Phuoc, Dien Dinh, and Hien T. Nguyen. "A character level based and word level based approach for Chinese-Vietnamese machine translation." *Computational intelligence and Neuroscience* 2016 (2016).

[47] Phan-Vu, Hong-Hai, Van-Nam Nguyen, Viet-Trung Tran, and Phan-Thuan Do. "Towards state-of-the-art English-Vietnamese neural machine translation." In *Proceedings of the Eighth International Symposium on Information and Communication Technology*, pp. 120-126. 2017.

[48] Jiang, Hao, Yue He, Mengfan Liao, Yanmei Jing, and Chao Zhang. "English-Vietnamese machine translation model based on sequence to sequence

algorithm." In *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pp. 1086-1091. IEEE, 2020.

[49] Le, Lac Si, Dang Van Thin, Ngan Luu-Thuy Nguyen, and Son Quoc Trinh. "A Multi-filter BiLSTM-CNN Architecture for Vietnamese Sentiment Analysis." In *International Conference on Computational Collective Intelligence*, pp. 752-763. Springer, Cham, 2020.

[50] Huang, Yong, Siwei Liu, Liangdong Qu, and Yongsheng Li. "Effective Vietnamese Sentiment Analysis Model Using Sentiment Word Embedding and Transfer Learning." In *International Conference of Pioneering Computer Scientists, Engineers and Educators*, pp. 36-46. Springer, Singapore, 2020.

[51] Lê, Ngoc C., Nguyen The Lam, Son Hong Nguyen, and Duc Thanh Nguyen. "On Vietnamese Sentiment Analysis: A Transfer Learning Method." In *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, pp. 1-5. IEEE, 2020.

[52] Vu, Dinh-Hong, and Anh-Cuong Le. "Topic-Guided RNN Model for Vietnamese Text Generation." In *Research in Intelligent and Computing in Engineering*, pp. 827-834. Springer, Singapore, 2021.

[53] Van Nguyen, Kiet, Duc-Vu Nguyen, Anh Gia-Tuan Nguyen, and Ngan Luu-Thuy Nguyen. "A Vietnamese Dataset for Evaluating Machine Reading Comprehension." *arXiv preprint arXiv:2009.14725* (2020).

[54] Lam, Quan Hoang, Quang Duy Le, Van Kiet Nguyen, and Ngan Luu-Thuy Nguyen. "UIT-ViIC: A Dataset for the First Evaluation on Vietnamese Image Captioning." In *International Conference on Computational Collective Intelligence*, pp. 730-742. Springer, Cham, 2020.

[55] Tran, Trung-Hieu, Long Phan, and Truong-Son Nguyen. "Leveraging Transfer Learning for Reliable Intelligence Identification on Vietnamese SNSs (ReINTEL)." *arXiv preprint arXiv:2012.07557* (2020).

[56] Nguyen, Anh Tuan, Mai Hoang Dao, and Dat Quoc Nguyen. "A Pilot Study of Text-to-SQL Semantic Parsing for Vietnamese." *arXiv preprint arXiv:2010.01891* (2020).

[57] Dat Quoc Nguyen, , Dai Quoc Nguyen, Thanh Vu, Mark Dras, and Mark Johnson. "A Fast and Accurate Vietnamese Word Segmenter." . In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)* (pp. 2582–2587).2018.

[58] Luu, T. A., and Y, Kazuhide. "Ứng dụng phương pháp Pointwise vào bài toán tách từ cho tiếng Việt.." . In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)* (pp. 2582–2587).2012.

[59] T. P. Nguyen, , and A. C. Le. "A hybrid approach to Vietnamese word segmentation." . In *2016 IEEE RIVF International Conference on Computing Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)* (pp. 114-119).2016.

[60] Thang, , and V. X., Luong. "Word segmentation of Vietnamese texts : a comparison of approaches.." . In *In Proceedings of the 6th International Conference on Language Resources and Evaluation* (pp. 114-119).2008.

[61] Mingxing Tan, , and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." (2020).